Issue 3, 1994

Personal Software

OS/2 Systems

LAN Systems

Random Data

Personal Software Magazine

IBM®

IBM Personal Software Magazine is published by IBM Personal Software Products User Group Relations, Austin, Texas, U.S.A.

User Group Relations Program Manager — Gene Barlow
Editorial Services — MDE Associates
Illustrations — Jamison American Originals
Typesetting — Dove Oaks Publishing

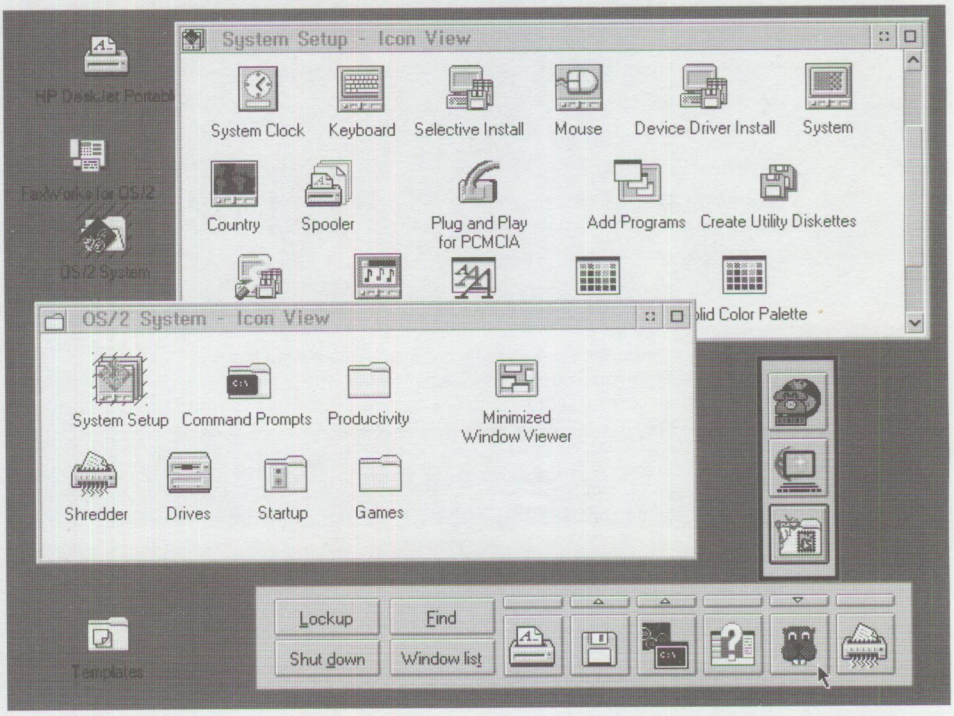
This publication is distributed only in bulk to PC and OS/2 user groups. Individual copies and subscriptions are not available.

Correspondence should be addressed to Gene Barlow, IBM Corporation, P.O. Box 201449, Austin TX 78720-1449. If you're electronically connected, you can send an Internet note to IBMPCUG@VNET.IBM.COM. IBM employees can send to IBMPCUG at AUSVM1.

What's New in OS/2 Warp

Adam Jollans
IBM Europe Personal Systems Marketing
Basingstoke, UK

Based on customer feedback and requirements, OS/2 Warp includes numerous enhancements to performance, the Workplace Shell*, installation, application support, and hardware device drivers. And, for the first time, the new OS/2 Warp comes with a BonusPak of more than ten 32-bit applications based on IBM ISV software products, and including Internet access. This article gives an overview of the new features in OS/2 Warp and its BonusPak of applications.*



Welcome to OS/2 Warp, IBM's latest and greatest 32-bit, pre-emptive multitasking operating system! Let's take a tour through OS/2 Warp, pointing out what is new in both the base system and the BonusPak of applications.

Performance Enhancements

- *Improved performance,* especially on low-memory computers, starting with as little as 4 MB. This improvement was achieved mainly by reducing the memory working-set requirements of OS/2 itself, by optimizing the kernel, and by combining DLLs so they load faster.

  Reducing the memory needed by OS/2 means there is more memory available to applications, so they run faster. The improved speed is most noticeable in low-memory computers, although there will be some performance improvements on all microcomputers.

  The OS/2 development laboratory's performance target was to achieve the same performance for OS/2 Warp running in 4 MB as for OS/2 2.1 running in 6 MB. Depending on application requirements, more than 4 MB of memory may be needed.

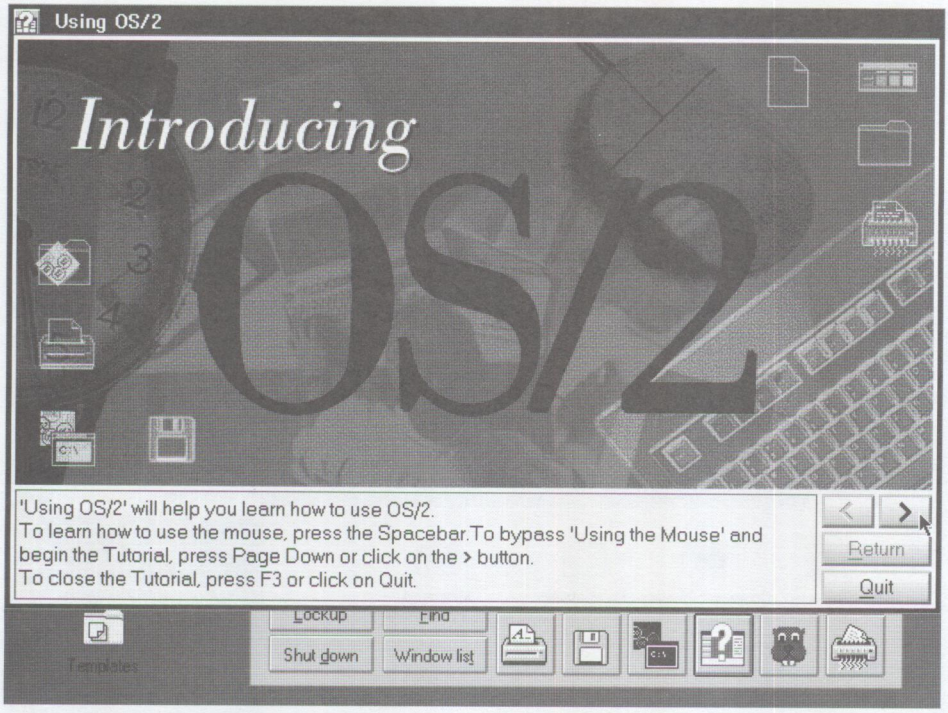
- *Fast-load option* for WIN-OS/2*. This option can be selected from the WIN-OS/2 Setup object in the System Setup folder. Selecting this option preloads the WIN-OS/2 subsystem when OS/2 Warp initializes, thus reducing the time to start a Windows** application running in the common WIN-OS/2 session.

- *Fully 32-bit Presentation Manager*.* The remaining components of Presentation Manager, including the windowing and print subsystems, have been converted to 32-bit code, which improves performance because there is no thunking (calling between 16-bit and 32-bit code) within PM. Thunking carries a heavy overhead for each procedure call because of the address and data conversions that must be performed for each thunk.

Workplace Shell Enhancements

- *LaunchPad.* The LaunchPad provides fast, one-click access to commonly used applications and functions. The icons on the LaunchPad are shadows of other Workplace Shell objects, and can be customized by the user. The appearance of the LaunchPad can also be customized.

- *New icons.* Three-dimensional icons are provided for the standard Workplace Shell objects. In addition, some icons, such as folders, change appearance when they are opened.

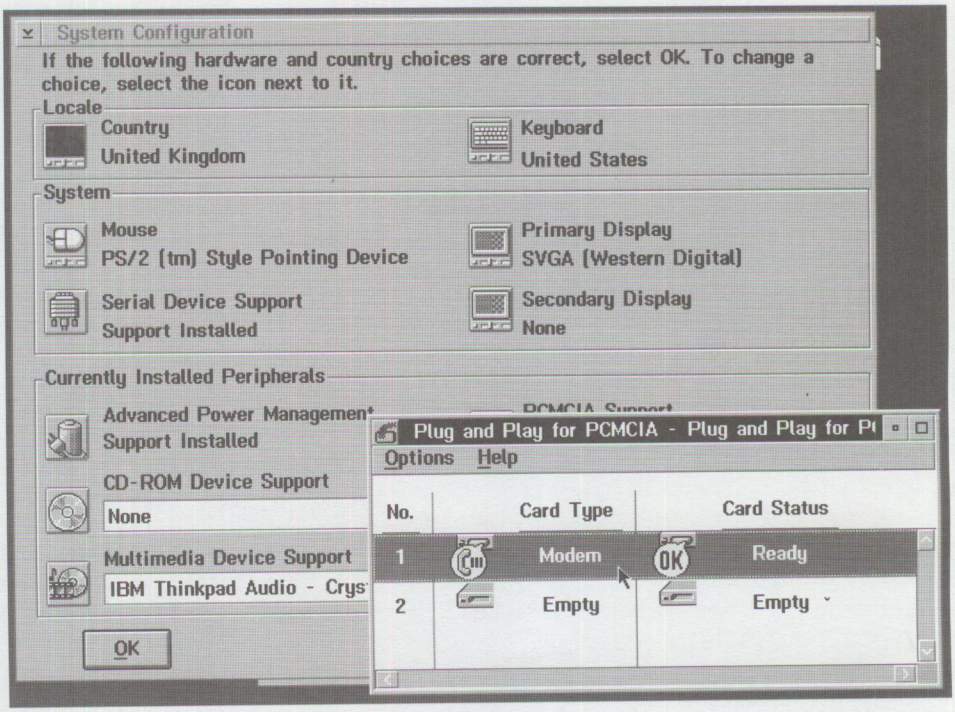


- *Improved Color and Scheme palettes.* The Scheme Palette now provides a standard choice of 24 color schemes; additional schemes can be customized by the user. There are now two color palettes: a Solid Color Palette with 16 colors, and a Mixed Color Palette with 256 colors.
- *Comet Cursor.* This enables a comet tail for the cursor, and is especially useful on laptops where it can be hard to find a lost cursor. The size of the trail increases with the speed of movement of the cursor, and is also customizable.
- *New options on pop-up menus.* Some enhancements have been made to the pop-up menus for Workplace Shell objects:
  - The Settings choice is now a separate option (it used to be in the Open menu)
  - *Open Parent* is now provided in the menus for open folders. Selecting it opens the parent of the folder, which is typically another folder.
  - *Pickup and Drop,* also called *Lazy Drag,* enables drag-and-drop without needing to hold the mouse button down during the drag. Both options are in the pop-up menu of the object.
- Simplified *Find* utility. The Find utility, available on pop-up menus or directly from the LaunchPad, is simplified for casual users. The search now defaults to searching on the object name (including partial searching with wild cards). The powerful search facilities are still available.
- *Automatic folder closing* prevents the Desktop from becoming cluttered with hierarchies of open folders, by enabling a folder to automatically close when a child folder or object is opened. It can be selected via the second page of the Window tab of the Settings notebook. (The Open Parent option in the pop-up menu of open folders can be used to retrace steps if necessary.)
- *Portable and recoverable Desktop.* The Desktop and key OS/2 configuration files can now be automatically saved each time OS/2 is started, and the archive can be used to recover from a damaged Desktop. Up to three archives can be kept. Because taking an archive subsequently increases the time it takes OS/2 to start up, archiving should normally be done only after major changes to the Desktop. Automatic archiving is set on by using the Archive tab of the Desktop's Settings notebook. The Desktop can be recovered by pressing Alt+F1 during OS/2 startup (while the white block is displayed in the top left corner of the screen).
- New *Tutorial.* OS/2 Warp comes with a completely rewritten Tutorial to help new users become familiar with using OS/2 Warp. The Tutorial is held in the Information folder, and is automatically started after OS/2 installation.

Installation Enhancements

- *Easy Installation,* the default option at installation time, enables a fast, almost one-button install of OS/2 Warp onto the computer's C drive. Easy Installation uses the automatic hardware detection (described below) to work out which device drivers to load and configure, and then presents the list to the user to verify. The user then needs to select the printer; everything else is automatic.
- *Advanced Installation,* the other option available at installation time, gives the user much more control over the installation. For example, the user can install OS/2 on drives other than the C drive; use the boot manager; use HPFS; and select the options to install. Like Easy Installation, Advanced Installation uses the automatic hardware detection to work out which device drivers to install and configure, and then presents the list to the user to verify.

- *Enhanced Selective Install* improves the user interface by using one-click graphics pushbuttons (instead of the previous two-stage check-box plus OK process) to review and select the hardware device drivers, including those for video, printer, and CD. Multiple printers can also be configured during Selective Install, either during or after initial installation.
- *Automatic hardware detection* at installation is substantially enhanced to include SVGA video adapters and improved detection of other hardware components.
- *Plug and Play for PCMCIA*** provides the ability to dynamically insert or remove PCMCIA cards while OS/2 is running. The appropriate PCMCIA device drivers are loaded dynamically, and the card can be used dynamically. Plug and Play is an enhanced version of the utility provided with IBM ThinkPad* computers. The addition of a wide variety of PCMCIA socket services has extended this support to many laptops and other PCMCIA systems.
- *Integrated multimedia installation.* Installation of MMPM/2 is now automatic and integrated into the OS/2 installation process, rather than being a separately initiated process.
- *Automatic Dual-Boot installation.* If OS/2 Warp is installed over an existing copy of DOS, then the Dual-Boot feature is installed automatically, and all the necessary changes to CONFIG.SYS and AUTOEXEC.BAT files are made.
- *New XDF format* for OS/2 diskettes. A new format for diskettes, called XDF, enables 1.88 MB of data to be stored on a 2 MB diskette (compared to the 1.44 MB standard format). Achieving 1.88 MB is done through more effective use of space on the diskette, rather than by any compression algorithm, and can be used in conjunction with compressed files. XDF is used for all the OS/2 system, device driver, and BonusPak diskettes – except for Disk 0 (the installation diskette) and Disk 1 – so the total number of diskettes is reduced. To read or write to XDF diskettes, the XDF device driver must be loaded.



- *Sharing of parallel port* for printer and CD-ROM. This sharing enables different devices, such as a printer and a CD-ROM drive, to be attached in turn to the parallel port and accessed from the appropriate device driver without rebooting the system.

Application Support Enhancements

- *Win32s application support* is included in OS/2 Warp. Win32s applications are a small group of advanced Windows applications that use the 32-bit memory capabilities of the Win32s API and can run under Windows 3.1.
- *SOM2* is included. The second generation of the System Object Model (SOM2) has been incorporated into OS/2 Warp, and has been used for the Workplace Shell. The major advantage of SOM2 is that it can be used across processes. This means that Workplace Shell-exploiting applications can be protected and run in address spaces different from the address space of the Workplace Shell, while still using all the Workplace Shell APIs via SOM2. The resulting benefit is that, if a Workplace Shell application goes wrong, it cannot crash the Workplace Shell's Desktop as well.
- Additional multimedia formats are now supported by OS/2 Warp through inclusion of the appropriate IOProcs. These formats include:
  - *INDEO*** *3.2*, the new level for Intel** software-motion video format
  - *MPEG*, an additional motion video format (using appropriate MPEG hardware)
  - *FLC/FLI*, a format for animated images, developed by Autodesk**

Display device drivers
ATI Mach 8/32
ATI Mach 8/64
Cirrus 5426, 5428, 5429, 5430, 5434
IBM ThinkPad 750, 755
S3 864
Tseng ET4000 W32, W32i, W32p
Weitek P9000, P9100
Western Digital C24, C26, C27, C31
Western Digital C33, C34

Multimedia video device drivers
ReelMagic (MPEG)
WinTV-02

Audio device drivers
Aztech Sound Galaxy
Compaq System Sounds, Microsoft Sound System (AD1848)
Crystal Semi CS4231, CS4248
ESS 688
Sound Blaster AWE32

Printer drivers
Omni printer driver

SCSI adapter device drivers
Adaptec 2740, 2840VL
Adaptec 2940

PCMCIA socket services
Ambra 486 SO 425C
AST Bravo NB Color, PowerExec 4/25SL, 4/25SLC
Compaq Concerto
CompuAdd 425 TX
IBM ThinkPad 350, 500, 720, 750, 755
IBM PS/2 E
Matsushita
NCR Safari 3180
NEC Versa C, E
Panasonic CF-V11P
Toshiba T3600CT, T4500, T4600, T4700. T4800
Zenith Z-lite 425L
Zeos Colornote

CD-ROM device drivers
Philips LMS CM205
Philips LMS CM206
Sony CDU-535
IDE-connected CD-ROM drives, including
Mitsumi CRMC-FXN
Philips LMS CM-207
Sony CDU-55E
Wearnes CDC-110
other IDE connected CD-ROMs

Diskette filter device drivers
XDF (new diskette format allowing 1.88 MB of data on a 2 MB diskette. Currently used for OS/2 and BonusPak diskettes.)

Figure 1. New Device Drivers in OS/2 Warp

- *Automatic Settings* for DOS/Windows applications. When a new program object is created in the Workplace Shell, the Migration Database (the database of optimized Settings for specific applications) is scanned, and the Settings are configured appropriately.

Hardware Device Drivers

The new device drivers in OS/2 Warp are listed in Figure 1.

For a complete list of OS/2 device drivers, refer to the latest issue of the PCMTABLE package, available on CompuServe and bulletin boards.

OS/2 Warp includes a BonusPak of more than ten useful applications.

What's in the BonusPak?

OS/2 Warp includes a BonusPak of more than ten useful applications. With these applications, you can make immediate productive use of OS/2 Warp. The rest of this article discusses the contents of the BonusPak.

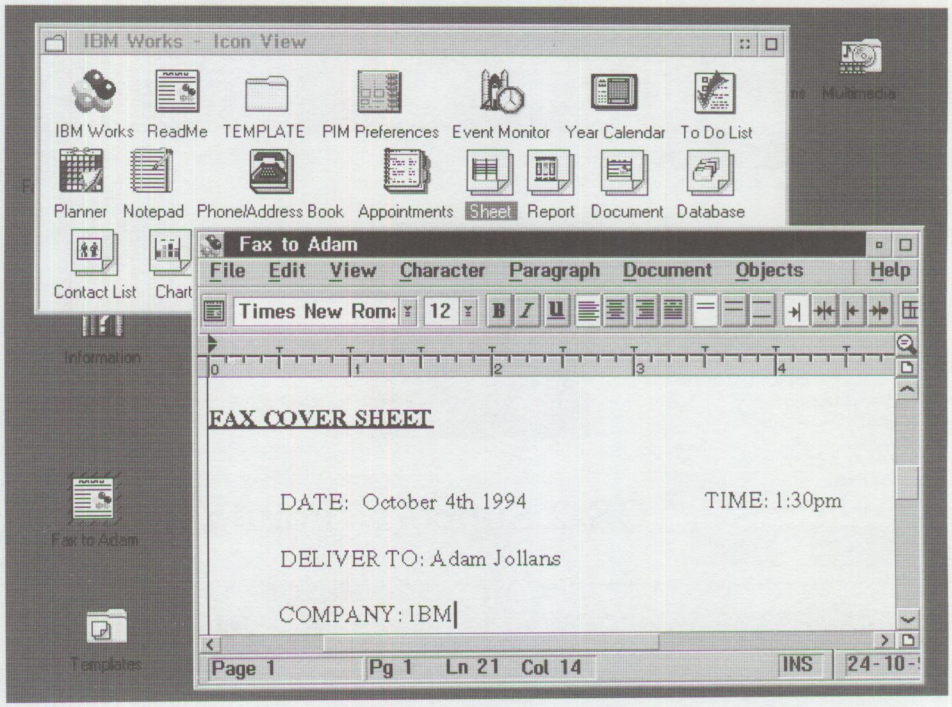
IBM Works: *IBM Works* is a suite of full-function 32-bit OS/2 applications, which are integrated with the Workplace Shell. For example, a new IBM Works document can be created by drag-drop from the Document template to the Desktop, word-processed by double-clicking to open the document, and

printed by drag-drop to a printer icon or fax icon.

The IBM Works included in the BonusPak is a close cousin of the IBM Works and Footprint Works software products, adding features such as Workplace Shell integration. IBM Works in the BonusPak is able to read files created by the previous release of IBM or Footprint Works.

Included in IBM Works are the following applications:

- A *word processor* that is able to create, modify, and print professional-quality documents using a range of fonts and including imbedded graphics. The word processor includes many advanced capabilities such as spell-checkers, mail-merge, table editor, style sheets, and keyboard macros, while remaining intuitively easy to use through features such as toolbars and an editable preview mode. The word processor can include static or hot-linked spreadsheets and charts created by other applications in IBM Works, or can be mail-merged with either the IBM Works database or the Personal Information Manager address book and contact lists.
- A *spreadsheet* that provides a 1024x768-cell spreadsheet, with financial, statistical, and other formula functions available. A variety of fonts and colors are available for each cell, and toolbars provide easy access to these and other functions. Multiple views of a spreadsheet are possible, and 2-D and 3-D charts can be generated and embedded in the spreadsheet.
- A *charting tool* enables graphs, pie charts, and other charts to be created and annotated based on spreadsheets or data entered directly. Graphics files can also be imported; drawing and text tools are available within the charting application; and charts can be linked into the word processor or the spreadsheet.

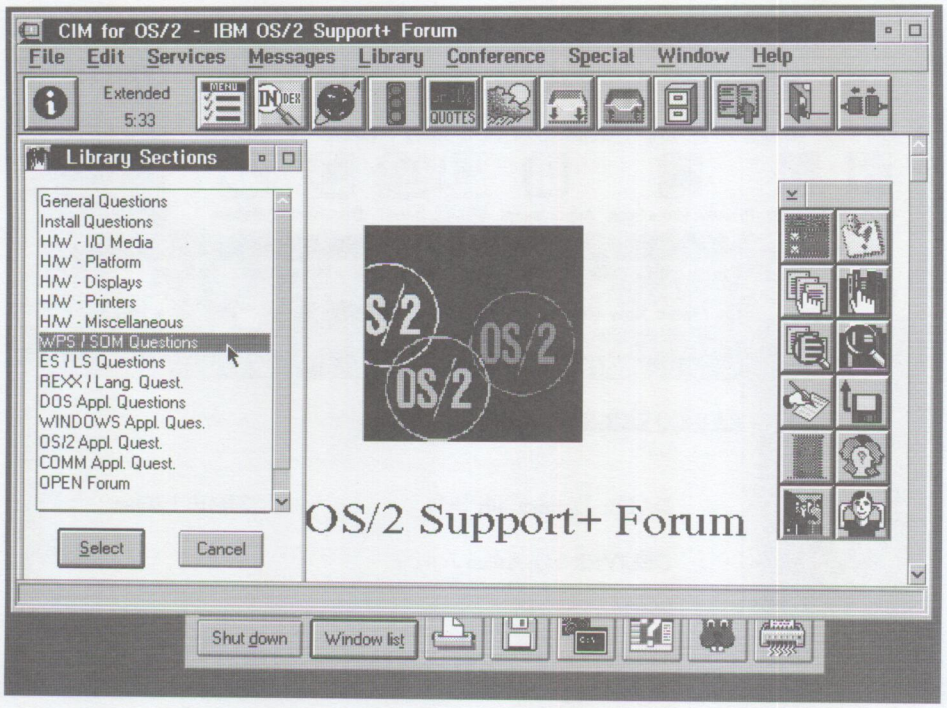


- A *database* provides both a screen design tool for creating online data-entry forms, and a data filer for storing and retrieving the information captured through these forms. Data can also be imported and exported from other database formats, and both text and graphics can be handled.
- A *report writer* enables printed and online reports to be generated based on the data in the database. Records can be selected, sorted, and presented using a variety of criteria. Both text and graphics fields can be handled, displayed, and printed.

Personal Information Manager: *Personal Information Manager (PIM)* is a suite of 32-bit OS/2 applications for managing diaries, calendars, address books, and to-do lists. The PIM applications exploit the drag-drop capabilities of the Workplace Shell; for example, dragging a contact from the address book to the appointment calendar automatically adds a meeting with that person on the appropriate day, and dragging a contact onto a document that contains merge fields will insert the contact details into the document and also attach the contact's fax number for subsequent faxing of the document.

Included in PIM are the following applications:

- *Appointment Book* provides an electronic diary where events can be scheduled and displayed in a variety of formats, such as on a weekly calendar. Recurring and overlapping appointments can be scheduled, moved, or copied using drag-and-drop; automatically created by dragging a contact from the address book onto the appointment book; and attached to an alarm to either alert the user or even launch other PIM applications. The appointment book can also be printed in a variety of formats.
- *Monthly Planner* provides a month's display on a grid format, linked to entries in the appointment book. Events can be rescheduled by drag-drop from the appointment book to a specific day in the monthly planner, or automatically created by dragging a contact onto the planner.



- *Yearly Calendar* provides a year's view of appointments, similar to the monthly planner, and also linked to the appointment book.
- *To-Do List* provides a way of recording outstanding actions, setting a priority, and attaching them to a specific day. The To-Do List can be searched on text or priorities.
- *Address Book* enables details of contacts to be recorded, including name, address, and multiple phone numbers. Subsets of the address book, called Contact Lists, can be created; these lists are automatically kept in step with the main address book. Contacts can then be used in conjunction with the PIM to schedule appointments; in conjunction with the IBM Works word processor for merged letters; or simply as a way of tracking phone calls and letters.
- *Notepad* provides an advanced electronic notepad that can store both text and graphics, be indexed with chapter headings, and be rearranged using drag-and-drop.
- *Preferences Notebook* can be used to customize the PIM settings, and the Event Monitor is automatically launched when the appointment book is accessed, to handle alarms attached to appointments.

FaxWorks: *FaxWorks* is a faxing application that enables multi-page faxes to be sent simply by printing from an application to a fax printer. Drag-and-drop of files onto the Fax icon can also be used for many files, such as text files and IBM Works files.

FaxWorks can also be used to receive faxes and store them on the hard disk of the computer. Received faxes can also be printed. FaxWorks can be used with both Class 1 and Class 2 faxes. FaxWorks is a subset of the FaxWorks** product, which also includes the ability to edit stored faxes.

HyperACCESS Lite: *HyperACCESS** *Lite* is an advanced ASCII terminal emulator program that can connect to a wide variety of bulletin boards. The object-oriented user interface of HyperACCESS enables individual icons to be set up for each bulletin board, enabling connection to the BBS to be made quickly and easily.

Multimedia Viewer/2: *Multimedia Viewer/2* provides a way to view a wide variety of multimedia files (such as bitmaps, Kodak** PhotoCD files, or audio files). It also enables users to organize and browse these multimedia files via a light-table, which displays "thumbnails" – miniature pictures of the contents of multimedia files. The Multimedia Viewer/2 included in the BonusPak is a subset of the Ultimedia Workplace/2 product.

Video IN: *Video IN* can be used to create software-motion videos from bitmaps or animation files. With a standard video-capture adapter, Video IN can also be used to create software-motion videos using input from a camcorder, VCR, or laserdisc player.

System Information Tool: The *System Information Tool* is a utility application to display details about the hardware currently installed in the computer; for example, the type of disk adapter, processor type and speed, or number of memory chips installed.

CompuServe Information Manager for OS/2: *CIM-OS/2* provides a graphical front-end for CompuServe**, enabling easy and fast access to the range of online news, information, and computing services provided through CompuServe. Also included are the files necessary for registering with CompuServe and starting an account.

Person to Person/2: *Person to Person/2** provides tools for sharing information and working collaboratively in real-time between two or more computer users, connected by phone lines.

Information can be shared via:

- common clipboards

- a shared chalkboard, enabling text and images to be viewed and annotated
- a chat-line
- desktop video conferencing (with appropriate hardware)

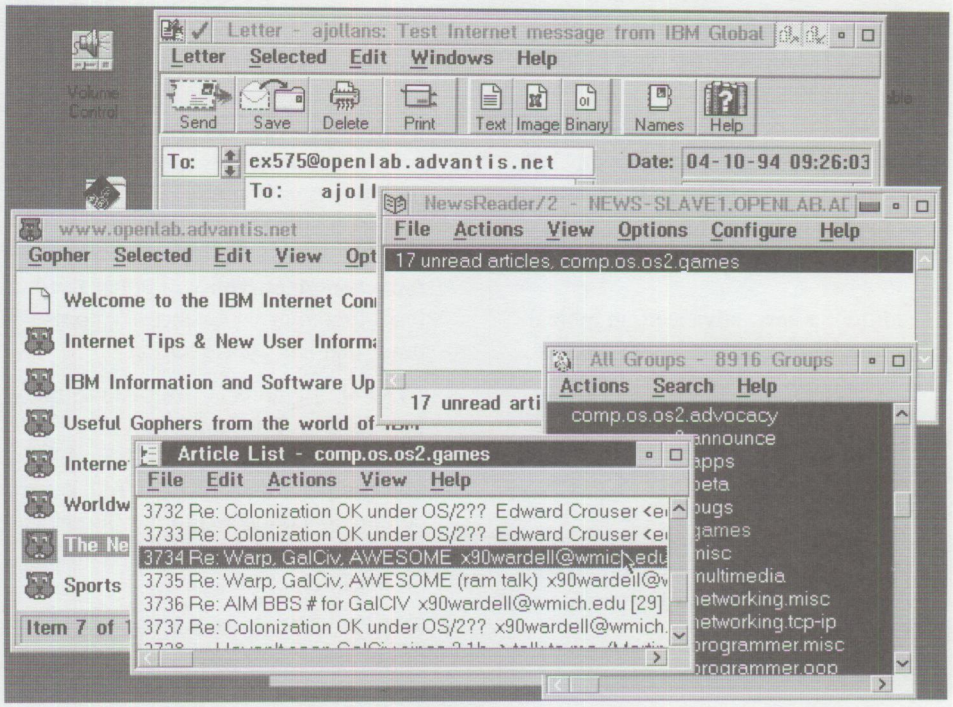
Person to Person/2 included in the BonusPak is a subset of the IBM Person to Person/2 product, which includes local-area network (LAN) and wide-area network (WAN) connectivity.

Internet Connection for OS/2:
Internet is a network connecting many thousands of computers and millions of users worldwide – including universities, businesses, government departments, and individuals. The Internet enables all these users to acquire, distribute, and share information, which includes simple text files, advanced multimedia files, and electronic-mail messages.

The Internet Connection for OS/2, included in the BonusPak, is a suite of 32-bit OS/2 applications enabling users to access the Internet through an appropriate service provider, and then to browse, explore, and "surf" the net. This can all be done over a phone line via a modem.

IBM, through its IBM Global Network (Advantis**), is one such service provider, and will be providing a number of worldwide access points during 4Q94. These will provide charged access to the Internet. In addition, the Internet Connection for OS/2 applications can also be used with other service providers.

The applications included in the Internet Connection for OS/2 can be divided into three groups: those for registration and access; easy-to-use applications that hide the complexities of the Internet; and low-level applications for Internet experts. The Internet contains so much information, on so many different computers worldwide, that it can be difficult to know what to look for, and where.



Applications such as Gopher** and the WebExplorer solve this problem, enabling any user to easily explore the Internet.

Registration and access applications include:

- *IBM Internet Registration*, which enables fast and easy registration onto the Internet via the IBM Global Network. A series of menus leads the user to enter user and credit card information; this information is then sent automatically to a registration server computer, which then returns the new userid and password.
- *IBM Internet Dialer*, which provides a fast and easy way of connecting to the Internet once registered, through the IBM Global Network.
- *Dial Other Internet Providers*, which enables access to the Internet using other service providers. The Internet Connection for OS/2 applications can be used in conjunction with any service provider.

Internet access is also automatically launched if one of the Internet Connection applications is started.

Easy-to-use Internet applications include:

- *Gopher,* a text-based tool for exploring the Internet, and for providing easy access to files and other facilities. Gopher provides a menu system across the computers on the Internet, jumping from one computer to another to provide the next level of menu as needed. Gopher also provides easy file retrieval or terminal access in conjunction with other computers on the Internet, once they have been reached via the menu system.
- *Ultimedia Mail/2 "Lite,"* which enables the user to send and receive electronic mail, to and from any other Internet user. This electronic mail is multimedia-enabled, and can include images, sound, and video as well as text. It thus provides a very powerful way to communicate with other companies, organizations, and individuals. Ultimedia Mail/2 provides a graphical

front-end for sending and receiving mail, and also facilities for storing and retrieving old mail.

- *NewsReader/2*, which provides a way of reading and automatically subscribing to any of the thousands of general- and special-interest forums on the Internet.

In addition, a new, advanced, graphical tool – the *WebExplorer** – will be provided soon (via Gopher). The WebExplorer provides the front-end to hyperlinked multimedia documents across the Internet – for example, starting at the IBM Home Page – thus hiding the complexities of Internet from the user. WebExplorer is the next-generation tool (beyond the menu-based Gopher) for exploring the Internet.

Low-level Internet applications include:

- *TelnetPM* and *TN3270*. These facilities provide remote logon – the ability to log on to other computers connected to the Internet. TelnetPM provides ASCII and VT100 terminal access, while TN3270 provides 3270 terminal access. To log on to a remote computer, you will either need a local logon, or try to access a computer that lets guests log on.
- *FTPPM*, a graphical front-end to the popular File Transfer Protocol (FTP) capability of the Internet. Using FTP, you can get copies of files from other computers connected to the Internet, as long as you know which computer on the Internet to access.
- *TCP/IP SLIP*, the underlying network protocol for connecting into the Internet over a phone line using a modem connected to the serial port of a computer. SLIP is automatically started and configured; you shouldn't need to be concerned about it.

Adam Jollans *works in IBM Europe/Middle East/Africa Personal Software Marketing in Basingstoke, UK, where he is responsible for Workplace product marketing. He has wide experience in the computer industry, including application design and programming, systems engineer working directly with large-account customers, and an assignment to the International Technical Support Organization at IBM Boca Raton, Florida.*

Adam joined IBM in 1984. He has an MA in Computer Science from Cambridge University, England, and is a Member of the British Computer Society. He was also co-editor of the OS/2 2.11 Power Techniques *redbook, published by QUE. He can be contacted via Internet at ajollans@vnet.ibm.com.*

OS/2 Warp, Version 3 – Details

Enhanced performance

- Runs swiftly and reliably with as little as 4 MB of memory
- Provides screen response times faster than before

Enhanced usability

- Allows you to swap PCMCIA cards without having to reconfigure and reboot your computer, through Plug and Play feature
- Lets you access frequently used functions with a single mouse click from the LaunchPad
- Includes the System Object Model V2.0 – a boon for software developers writing new object applications
- Saves multiple steps with one-button load of your applications from the Workplace Shell
- Provides a new online tutorial that lets you perform actual tasks while you learn
- Saves multiple steps with one-button load of your applications from the Workplace Shell
- Presents system instructions and error messages in easy-to-understand language
- Permits you to drag and drop without holding down the mouse button – helpful for notebook users
- Allows you to seamlessly share your parallel port between your printer and CD-ROM – no reconfiguring necessary
- Lets you group all files for a project in a work area – then open, close, or hide them all at once with a click of the mouse

Workplace Shell

- Is easy and fun to use, with 3-D and animated icons plus new comet cursor
- Works the way you do – provides a LaunchPad that can be customized with any of your most-used functions
- Offers simple-to-use iconic drag-and-drop capabilities that are consistent across the operating system
- Allows network resources, such as folders and printers, to be accessed as objects on your desktop
- Lets you create multiple desktops – great for families – then allows you to use the same desktop on your portable computer
- Provides a single, easy approach to managing multiple system resources – printers, drives, files – and applications
- Shields you from complex computer functions.
- Boosts productivity for experienced computer users; reduces training requirements for novices

BonusPak for OS/2 Warp

- Ships with OS/2 Warp and with OS/2 Warp LAN Client – a separate product that gives you outstanding value
- Features a set of full-function applications – word processing, spreadsheet, charting, data filing, and report writing – all with one easy-to-learn graphical interface
- Provides fax and communication software, including HyperACCESS Lite for OS/2 – the easy way to access online services, bulletin boards, or other PCs and mainframes
- Includes Personal Information Manager, CompuServe Information Manager for OS/2, e-mail, and IBM Person to Person for OS/2 – software for collaborative computing
- Lets you view software motion video and images and play audio files with the Multimedia Viewer, or capture, clip, and play synchronized video and audio with Video IN for OS/2
- Offers the System Information Tool – a utility that assists with system and software problem resolution

Internet Connection

- Is part of BonusPak for OS/2 Warp – shipped at no extra charge with OS/2 Warp
- Provides a complete information-highway access solution
- Lets you seamlessly navigate through the Internet with easy-to-use graphical interfaces
- Gives you an automatic dial-up connection to IBM Internet Connection Services, or access via any SLIP-enabled provider
- Includes IBM WebExplorer (see Note 2), Gopher, Telnet, FTP, Sockets API, NewsReader, e-mail, and TCP/IP dial capabilities
- Allows you to travel in the information highway's fast lane, with high-speed modem throughput
- If you have access to the Internet, you can find additional information on IBM's World Wide Web server address at www.ibm.com.

Improved installation

- Saves you time with easy express install
- Asks few questions, so you can be up and running quickly
- Uses fewer diskettes
- Simplifies installation process with built-in hardware detection
- Provides automatic dual boot when DOS is present

Advanced application

- Supports DOS, Windows (see Note 1), and OS/2 support applications
- Enables you to run more applications from one desktop than any other operating system
- Allows you to run multiple Windows (see Note 1) applications on the desktop simultaneously
- Permits you to window and cut-and-paste between DOS applications

Preemptive multitasking

- Allows you to work in a more productive way, attending to one task while one or more others run in the background
- Enables execution of multiple applications at once – including multiple DOS, Windows (see Note 1), and OS/2 applications
- Provides the functionality of several computers in one

Multithreading

- Gives you the capability to complete many processes within an application simultaneously
- Provides better application response time

Crash Protection*

- Creates "virtual machines" for each application – designed to help keep your system running when one application fails
- Gives you proven stability for DOS and Windows (see Note 1) applications, while allowing them to exchange data with each other and with OS/2 applications
- Eliminates the need to reboot, reconnect, or reconfigure your computer if a particular application should fail

32-bit addressability

- Exploits today's microprocessing technology – Pentium**, 486, and 386 processors – with source portability to PowerPC*
- Provides a strong platform for advanced applications – especially those involving full-motion video, digital sound, speech, and handwriting recognition

Multimedia support

- Gives you enhanced audio, basic image support, and software motion video playback capabilities
- Allows automatic detection and driver configuration for multimedia system features at installation
- Lets you attach sounds to Workplace Shell actions and run movies

Enhanced memory

- Means processor-intensive applications run management faster; even Windows (see Note 1) and DOS programs can get a performance boost
- Recognizes and utilizes all available memory
- Opens the door to increased use of virtual memory, limited only by available disk space

Addressability

- 32-bit

System requirements

- i386 SX microprocessor (or compatible) or higher; VGA display (minimum); fax/modem (9600 bps or higher, for online access to the information highway)

Memory requirements

- 4 MB minimum

Disk-space requirements

- 35 MB to 50 MB of free hard-disk space, depending on the installation options selected
- BonusPak for OS/2 Warp requires up to 30 MB additional free space (user-selectable)

Notes:

(1) There are two OS/2 Warp products to choose from, depending on what your system currently has installed, and what types of applications you want to run. Both of these products include the support needed to run a wide variety of OS/2 and DOS applications. If you already have Windows installed, the OS/2 Warp product that uses your existing Windows is the product to choose. If you don't have Windows installed and want to run Windows applications, choose the OS/2 Warp product that includes IBM's WIN-OS/2 code, which provides the support required to run most Windows applications.

(2) IBM plans to make WebExplorer available to all users via download. Included in package after January 1995.

OS/2 Warp Versus Windows 95: A Decision-Maker's Guide to 32-Bit Operating-System Technology

Prepared by
IBM Personal Software Marketing
October 1994

Executive Summary

*This article tells PC users and decision-makers about the benefits of OS/2 and points out critical weaknesses in Microsoft's forthcoming Windows 95 operating system. At the heart of the discussion are key architectural, operational, and strategic flaws in the Windows 95 OS design and strategy – flaws that Microsoft** has either downplayed or ignored in its efforts to market Windows 95 as the next-generation Windows desktop platform.*

For example, you'll learn:

- *Why OS/2's ability to isolate individual 16-bit Windows applications into their own separate VDMs provides a level of inter-application protection that is unavailable under Windows 3.1 or Windows 95*
- *How this same isolation also allows OS/2 to preemptively multitask existing 16-bit Windows applications, with no impact on native application performance*
- *Why having a comprehensive System Object Model (SOM) is important, and how OS/2's SOM implementation acts as the "glue" to the Workplace Shell interface*
- *Ways in which OS/2's Virtual DOS Machine implementation is more flexible than Windows 95's.*

Major topics include:

- *Architectural flaws that may compromise Windows 95's stability when running 16-bit Windows applications*
- *How these same flaws also limit Windows 95's current multitasking capabilities with a mixture of application types*
- *Why the lack of a System Object Model makes the Windows 95 interface "fragile"*
- *Ways in which Windows 95's DOS heritage render the product inflexible when dealing with 16-bit DOS device drivers*

At the end of each section, a direct comparison is made between the Windows 95 implementation of a particular subsystem or feature/function, and that of the leader in 32-bit desktop operating systems, IBM's Operating System/2.*

The material in this article is based on an in-depth analysis of the following non-confidential, currently available sources:

- *Microsoft's public statements regarding Windows 95's design characteristics*
- *Various presentations given at trade shows by industry consultants*
- *The references cited at the end of the article.*

OS/2 Warp – The Right Solution

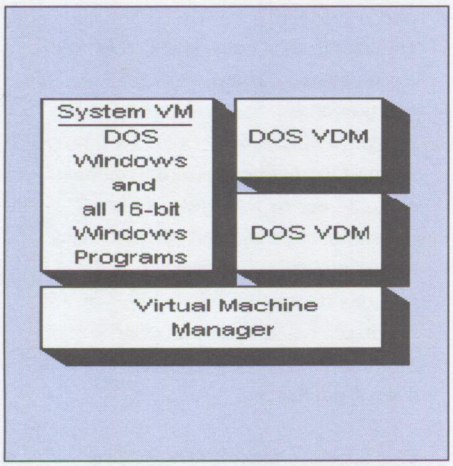
Choosing the right operating system … in many ways it's the most important personal-computer technology decision you'll make in this century. Choose wisely, and you'll reap the benefits for years. Choose poorly, and you may find yourself in a quagmire of under-performing software and inadequate computing power.

What constitutes a wise choice in today's confusing PC marketplace? Simple: the product that does the best job of preserving your existing investments while opening the door to the future. In a nutshell, the wise choice is Operating System/2.

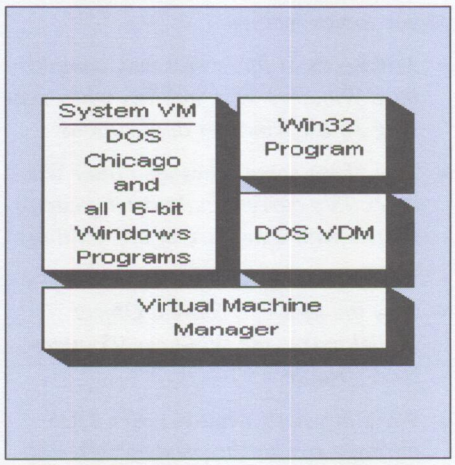
OS/2 – The World's Most Popular 32-Bit Operating System: Why OS/2? Because it represents the most logical upgrade path for today's PC users. OS/2 Warp preserves your investment in 16-bit DOS and Windows applications while providing access to a new world of 32-bit, object-oriented technology.

Upgrading to OS/2 is a win/win proposition. Just ask any of the more than five million OS/2 users – over eight times as many users as Microsoft's current 32-bit offering, Windows NT**. These users are people just like you who have outgrown their existing DOS or Windows environments and who are looking for more – more power, more functionality, more stability.

With OS/2, they've found a powerful mix of backward compatibility, 32-bit processing power, and ease of use, along with the kind of rock-solid reliability that only a mature, established operat-



Windows 3.1: The OS "House of Cards"



Windows 95: Same Code, Different Packaging

ing system platform can deliver. With the release of Version 3, OS/2 is entering its third generation, and the product's reputation for reliability and price/performance is unmatched in the PC industry.

But What About Windows 95?

This is the question that perplexes both corporate decision-makers and end users alike. With all of the media hype surrounding this next generation of Microsoft Windows, many customers feel paralyzed when making operating-system purchasing decisions. The fear of "missing out" on Windows 95 is overwhelming for some.

But as experience with the initial beta release of Windows 95 has demonstrated, Microsoft's "next generation" of Windows is far less compelling than they would lead you to believe. In fact, the core of Windows 95 is probably running on a PC near you – it's called Microsoft Windows 3.1.

Architecture

Windows 95 – Same Code, Different Packaging: "How can that be? It looks so different!"

Looks can be deceiving. While Windows 95 indeed sports a radically different user interface (more on that later), as you peel away the layers of GUI and packaging you'll discover a product that looks remarkably like Windows 3.1. In fact, Windows 95 retains so much of its original DOS/Windows heritage that it retains the latter's most notorious operational characteristic flaw: instability.

For example, under Windows 3.1, all applications, as well as the operating system code itself, share a single memory-address space. While such a memory-management model breeds performance, it also means that an error in any single application can potentially crash the entire operating system.

This crashing phenomenon is often referred to as a General Protection Fault, or GPF, and has been the bane of Windows users since version 3.0. It is because of this inherent architectural weakness that Windows 3.1 has gained a well-known reputation of being an unstable, unreliable operating environment.

Under Windows 95, this same single-address-space model (referred to as the "System Virtual Machine") is retained, along with the inherent weakness of leaving key portions of the operating system code exposed to potentially buggy applications. Thus, the same application failures that crashed Windows 3.1 can potentially bring down the entire Windows 95 operating system.

To their credit, Microsoft has made great strides in cleaning up many of the bugs in the original Windows 3.1 code while preparing it for inclusion with Windows 95. However, they cannot avoid the inherent architectural flaws that the Windows 3.1 Single System VM model introduces. There will always remain the possibility of an errant application causing a disastrous system crash.

OS/2 – Same Code, Better Implementation: OS/2 eliminates the Single System VM stability problem by letting you run Windows applications in their own separate sessions, or Virtual DOS Machines (VDMs). Thus, if an application fails under OS/2, the effect of the failure is limited to the individual session. Other applications, as well as the operating system itself, remain unaffected.

And by retaining much of the original Windows 3.1 code base, OS/2's environment remains highly backward-compatible with Windows 3.1 applications and device drivers.

Multitasking

Windows 95 – A "Semi-Preemptive" Task Switcher? One of Microsoft's biggest selling points for Windows 95 has been the promise of a new breed of 32-bit Windows applications. These applications are to be preemptively multitasked by the Windows 95 operating system, and will have access to advanced performance-enhancing techniques like multithreading. (OS/2 has offered preemptive multitasking and multithreading for years.)

Let's define the difference between preemptive and cooperative multitasking. *Preemption* is an involuntary loss of control that the application must handle. In *cooperative multitasking,* the application is given control, and it is the

application's responsibility to give up control so that other applications may execute.

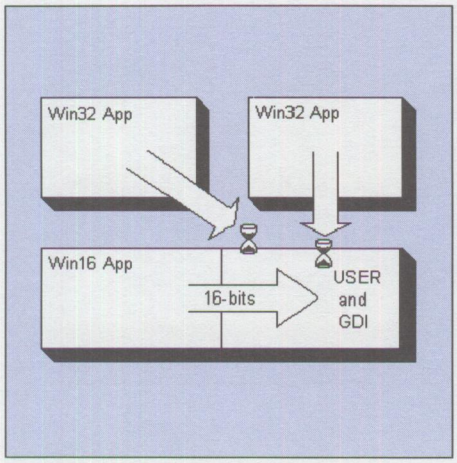
The move to a preemptive multitasking model represents a a significant departure from Windows 3.1. Under the Windows 3.1 environment, applications must "cooperate" in order for multitasking to occur. Each program must yield to the operating system so that the OS can switch control of the PC's processor to a different application. This is often referred to as *cooperative multitasking* or *task-switching*.

It is a well-known fact that the Windows cooperative multitasking model is inefficient. It also forces programmers to code their applications in a way that adds complexity and hinders performance. So it comes as no surprise that Microsoft's promise of preemptive multitasking in Windows 95 has been heralded as one of the new platform's most important features.

But Microsoft isn't telling the whole story about Windows 95's multitasking architecture. In reality, unless you work exclusively with 32-bit Win32 applications, you won't reap the benefits of true preemptive multitasking.

Why not? Because of Windows 95's heavy reliance on 16-bit, Windows 3.1-era code. Under Windows 95, both 16-bit and 32-bit applications rely on 16-bit code structures that reside within the System VM – code that has been brought over from Windows 3.1.

While the "bitness" of the code itself isn't significant, the environment from which it hails is. Windows 3.1 was written as a cooperative – not preemptive – multitasking environment. When you introduce portions of its code into a preemptive setting, where more than one task may be vying for its services at any given time, the code could break.



Windows 95: Roadblocks on the Way to Preemptive Multitasking

Regardless of the mixture of application types, OS/2 smoothly multitasks dozens of concurrent programs.

To safeguard against this sort of "code breakdown," Microsoft has serialized access to key portions of the Windows 95 infrastructure – most notably the USER (window management) and GDI (graphics device interface) subsystems. In technical terms, this is referred to as a *non-reentrant* design, meaning that only one application can execute within these modules at any given time.

While such an approach works with Win32 applications – which can be preempted at any point during their execution – it breaks down once a 16-bit Windows (Win16) application begins to execute. As it stands, currently shipping Win16 applications cannot be reliably preempted during execution. Attempting to do so while such an application is calling on a non-reentrant, 16-bit-code module can cause the entire operating system to crash.

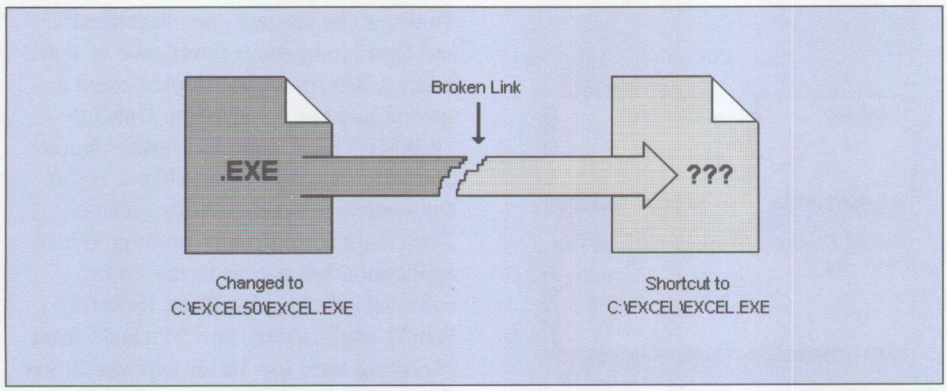
To avoid the scenario just described, and thus retain some semblance of multitasking, Microsoft has implemented a special locking mechanism. Dubbed "WinMUTEX," this mechanism denies access to the older code when a 16-bit application has called on its services. Thus, only the currently running Win16 application has access to the 16-bit code; all other applications, including Win32 applications, are "blocked" from executing until the 16-bit application has finished and the environment has been made safe for the next task.

In practice, the performance hit associated with this locking phenomenon is minimal when running 32-bit applications exclusively. However, when you introduce a mixture of 16- and 32-bit applications – the most likely scenario given the projected lack of available Win32 products – WinMUTEX becomes a major problem.

Most 16-bit Windows applications are notorious for failing to yield properly under Windows 3.1, and until they do so under Windows 95, all other applications will be blocked from accessing USER and/or GDI (in reality, only 50 percent of GDI calls are affected, but these are the most common functions, so the result is the same).

Taken as a whole, these two compromises – the serialization of subsystem access and WinMUTEX – create what would be best described as a "semi-preemptive" multitasking environment. And while the resulting hourglass is expected under a cooperatively multitasked environment, it seems out of place in a next-generation Windows that supposedly preemptively multitasks native Win32 applications.

OS/2 – True Preemption for Better Performance: OS/2 has featured true preemptive multitasking of native applications since day one. Regardless of the mixture of application types, OS/2 smoothly multitasks dozens of concurrent programs, and its reentrant subsys-



Windows 95: Links Are Easily Broken

tems enable it to service multiple concurrent requests without the overhead of a WinMUTEX implementation.

Thanks to its ability to run separate VDMs, OS/2 can also preemptively multitask existing 16-bit Windows applications, while Windows 95 cannot. Thus, you can have DOS, Windows, and OS/2 applications running concurrently, side-by-side, without any performance penalties, and all preemptively multitasked. This is a feature that Windows 95 seems to be unable to match without underlying architecture changes, and it is a welcome addition to any power-user's arsenal.

Interface

Windows 95 – Beauty That's Only Skin-Deep: Another major feature of Windows 95, one that has drawn considerable attention from the industry press, is its new user interface. Terms like "object-oriented" and "desktop metaphor" are often used to describe this radically different Windows look.

But, as with most of Windows 95's underpinnings, the actual foundation beneath the product's user interface is nothing more than an extension to what already existed in Windows 3.1. Unlike a true object-oriented environment, where links between individual objects are "live" and are updated automatically, the Windows 95 GUI is static. "Objects" on the Windows 95 desktop are merely pointers to files on the disk. Properties for these objects are stored in .INI files (for Windows applications) or .PIF files (for DOS applications), and links between them (called *shortcuts* under Windows 95) are equally static.

For example, if you create a shortcut to an executable file and place it on the Windows 95 desktop, then rename the original executable, the shortcut will essentially be severed. To re-establish it, you'll have to re-create the shortcut from scratch.

In a true object-oriented environment, all shortcut-like links to the executable would have been updated automatically by the underlying object-management model. Windows 95 has no such underpinnings, so links are easily broken by novice users who are unfamiliar with this weakness of the Windows 95 interface.

Going hand-in-hand with Windows 95's shortcut mechanism is the product's support for long file and directory names on FAT volumes. Microsoft is emphasizing Windows 95's ability to automatically convert long file and directory names into 8.3 character abbreviations for compatibility with existing DOS and Windows applications. What they seem to be ignoring, however, is the fact that promoting the use of long names can be disastrous when there is no underlying object model.

Take, for example, novice users who, upon discovering long filenames, decide to "reorganize" their hard disk. They gleefully rename directories at will, unaware that they are severing shortcut after shortcut in the process. Suddenly, none of their applications works, and technical support is called in to undo the damage (which in some cases may mean reinstalling both the operating system and applications).

The Windows 95 desktop itself is not an OLE 2.0 object. This statement in itself has no ramifications until you start understanding what type of integration is lost due to this lack of object technology. This deficiency in the product means that an application is not well integrated with the desktop, and does not inherit any of the advantages like drag-and-drop support.

Heralded by Microsoft as one of Windows 95's key selling points, the new Windows interface may in the end prove to be one of its biggest flaws. Without an underlying system object model to tie everything together, this new "shell" may prove to be a technical support headache.

OS/2 – True Object Orientation: OS/2's WorkPlace Shell is a true object-oriented interface. The underlying System Object Model (SOM) provides complete object-tracking, so that simple operations like dragging a directory to another directory won't invalidate links and other interface structures. This is easier on both novices and tech support staff alike.

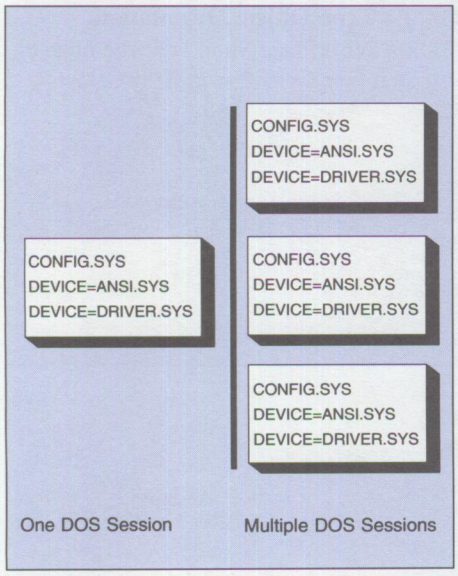
SOM also allows applications to fully manipulate the Workplace Shell interface. A good example is cc:Mail for OS/2, which uses SOM to seamlessly integrate its in/outbox interfaces with the Workplace Shell desktop. This level of integration is not possible under Windows 95, because its shell is itself not an object.

Application Support

Windows 95 – Still DOS After All These Years: "Windows 95 eliminates the need for DOS. It is a true operating system …"

This is one of the more colorful myths surrounding Microsoft's Windows 95 operating environment. Microsoft claims that Windows 95 eliminates the need for DOS – DOS and Windows are now completely integrated, and all the old restrictions that DOS brought to the table have been eliminated.

While it is true that you will no longer have to purchase a separate DOS product in order to install and use Windows 95, this in no way constitutes the eradication of DOS as a part of the Windows operating system equation. DOS is still there, lurking in the shadows. It's just been cleverly disguised by a different Windows GUI. And though much of its functionality, including file-system access, has been replaced by 32-bit Windows 95 VxDs (Virtual Device Drivers), there are still ways in which DOS can hinder the Windows environment.



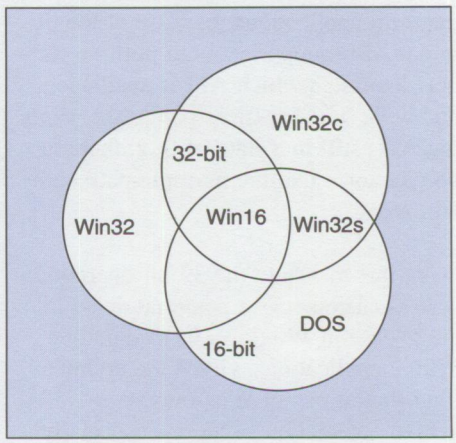
Windows 95 Loads All the Device Drivers for Each DOS Session in Conventional Memory

Take real-mode device drivers, for example. Under DOS/Windows 3.1, you were forced to load all DOS device drivers at DOS boot time via the CONFIG.SYS file. These drivers then occupied all DOS sessions under Windows' 386 Enhanced Mode, impacting their available conventional memory, and limiting the overall configurability of the Windows VDM architecture.

Windows 95 suffers from this very same limitation. Any real-mode DOS device drivers that you want to access from within Windows 95 must be loaded via CONFIG.SYS at boot time. Thus, if you want access to a particular resource, and this resource requires a DOS device driver, you'll be forced to pay a penalty in terms of lost conventional memory and potential compatibility problems across all Windows 95 VDMs.

And what about troublesome applications like games? Windows 95 features a special DOS session – the "Single MS-DOS** Application Mode" – that allows such applications to execute unencumbered by the confines of a traditional Virtual DOS Machine (virtual I/O, video memory, etc.). What Microsoft doesn't publicize, however, is that in order to invoke this mode, you must essentially shut down Windows 95. All running applications close, and the Windows 95 GUI itself is paged to disk. This entire process can take up to a minute, depending on the speed of the hardware in use and the number of open applications – quite a disruption, especially when you're trying to finish that last-minute memo or download a large file from a host system.

OS/2 Runs DOS Applications Better than DOS: OS/2 really does eliminate the need for DOS. Its VDMs are completely configurable, allowing you to create individual CONFIG.SYS and AUTOEXEC.BAT files for each DOS session. This option is important in those situations where a single device driver or TSR configuration for all VDMs would be inadequate.



Microsoft's API Quagmire

OS/2's VDMs are also highly backward-compatible, and can also be configured to allow direct hardware access for applications that require it. And if an application truly refuses to run under OS/2, you can use the dual-boot option to run real DOS in about the same amount of time it takes you to invoke Windows 95's "Single MS-DOS Application Mode."

Independent Software Vendor Commitments

Windows 95 – An ISV Headache: One area where Microsoft continues to be uncertain is the subject of API standards. Independent Software Vendors (ISVs) have been fighting an uphill battle in their efforts to pin down Microsoft's overall API strategy. This is especially true of the native Windows 95 API, Win32c, which is itself a subset of the full Win32 API published nearly two years ago and implemented on Windows NT.

Further exacerbating the situation is Microsoft's continual updating of the Win32c specification. New APIs emerge almost monthly, many of which extend Win32 in ways that tie applications to the Windows 95 platform. This has aggravated ISVs who want to write cross-platform applications for Windows, Windows NT, and Windows 95. The only way these ISVs can write cross-

platform applications, because of the different APIs supported, is to poll the Kernel, determine which API is available, and write dual- or triple-path code. With the APIs still in a state of flux, there is no guarantee that the multiple-path code will work.

What this means to the 32-bit operating system customer is a potential delay in the release of Windows 95-compatible Win32 applications. Given the architectural limitations of Windows 95's Win16 application support – especially when multitasking and stability are major considerations – lack of Win32 applications could represent a serious obstacle to the platform's widespread adoption. Windows 95 needs Win32 applications before it even begins to make sense as a replacement for Windows 3.1. But given the confusion and frustration in the ISV community, it may be some time before we see a substantial selection of Win32 titles.

OS/2 – A Consistent Message: In contrast to Microsoft's "API du jour" strategy, IBM has stood firm on its promises to support open standards and honor ISV commitments. There is one 32-bit OS/2 Presentation Manager API for both client and server systems. Application functions written to that API will work across OS/2 versions running on Intel-based PCs, and will be easily portable to more advanced implementations in the future (including OS/2 for PowerPC).

OS/2 currently boasts over 2000 native applications, all of which tap into its superior multitasking and performance.

OS/2 – The Right Answer

As you can see, so far Microsoft's Windows 95 operating system is long on hype and somewhat short on technology. If you've followed their product offerings over the past few years, this revelation should come as no surprise. Microsoft has a track record of delivering "cosmetically advanced" operating systems while ignoring the more impor-

OS/2 Warp vs. Windows 95 - Architecture		
Feature	**OS/2 Warp**	**Windows 95**
32-Bit Window Management	Yes	No [1]
32-Bit Graphics Subsystem	Yes	No [2]
32-Bit Printing Subsystem	Yes	Yes
32-Bit Multimedia Subsystem	Yes	Yes
32-Bit Kernel	Yes	Yes
Demand-Paged Virtual Memory	Yes	Yes
HPFS Support	Yes	No
Non-Locking Input Queue [3] (applications can keep running)	Yes	No

[1] USER is 16-bit, non-reentrant code
[2] 50 percent of GDI calls are serviced by 16-bit, non-reentrant code
[3] OS/2 Warp has an engine that will unlock the input queue if it is locked

OS/2 Warp vs. Windows 95 - Application Environments		
Feature	**OS/2 Warp**	**Windows 95**
16-Bit OS/2 PM Applications	Yes	No
32-Bit OS/2 PM Applications	Yes	No
Win32s Applications (Ver 1.0 & 1.1)	Yes	Yes
Preemptive Multitasking [4]	Yes	No
Win16 Application Support	Yes	Yes
Win16 Device Driver Support	Yes	Some [5]
Number of 32-bit Applications Available	2000+	0 (zero) [6]

[4] See chart on multitasking comparison
[5] Windows 3.x communications drivers need to be re-written
[6] Native Windows 95 applications

tant issues like robustness, capacity, and true object orientation. Windows 95 looks more and more like a warmed-over version of yesterday's technology, not the next-generation Windows platform that Microsoft is advertising it to be.

In contrast, IBM has a very different track record, one that speaks of commitment to open standards and listening to customer needs. This is the same company that has been developing cutting-edge OS technology for mainframe and minicomputer systems since the dawn of the information age. With OS/2, IBM has laid the foundation for a truly robust, high-capacity computing environment that preserves your existing investments while opening the door to the future.

You can see the difference in areas like the OS/2 user interface. The Workplace Shell, in conjunction with the System Object Model (SOM), provides a truly object-oriented computing environment, one that thinks for you and doesn't break down when you try to tap into its power. Likewise, OS/2's multitasking represents a no-compromises approach to bringing this powerful capability to the masses. From native OS/2 applications to its robust WIN-OS/2 VDMs, it is an operating system that can juggle your most complex tasks with ease.

The wise choice is obvious: OS/2 Warp has the backward compatibility you want, the stability and reliability you need, and the kind of rock-solid commitment to excellence you've come to expect from the world's premier software company – IBM.

To GET WARPed, call 1-800-IBM-CALL, or see your local software dealer.

OS/2 Warp vs. Windows 95 – Multitasking Characteristics

Feature	OS/2 Warp	Windows 95
Preemption of 32-bit OS/2 and WIN32s Version 1.1 Applications	Yes	Yes
Preemption of DOS Applications	Yes	Yes
Preemption of Win16 Applications	Yes	No
Preemption of Mixed 16/32-bit Applications [7]	Yes	No [8]
Multiple, Protected Win16 VDMs	Yes	No [9]
Crash Protection	Yes	No [10]
Preemptive Multi-Threading	Yes	Yes

[7] 16- and 32-bit OS/2, WIN16, and WIN32s v1.1 applications

[8] WinMUTEX prohibits access to USER and portions of GDI when a Win16 application is executing

[9] All 16-bit applications share a single address space, the System Virtual Machine (VM)

[10] Key operating-system code structures (USER and GDI) share the System VM address space with 16-bit applications

OS/2 Warp vs. Windows 95 – User Interface

Feature	OS/2 Warp	Windows 95
Folder Work Areas	Yes	No
Integration with Operating SOM	Yes	No [11]
LaunchPad	Yes	Yes
Drag-and-Drop Deletion	Yes	No
Drag-and-Drop Faxing	Yes	Yes
Drag-and-Drop Access Paths (change execution paths and it still works)	Yes	No
Object-Type Templates	Yes	No
Parent Folder Closing Options	Yes	No

[11] Windows 95 shell components are not OLE 2.01 "objects"

References:

King, Adrian, "Windows, the Next Generation: An Advanced Look at the Architecture of Chicago," *Microsoft Systems Journal*, January 1994, pp. 15-24.

King, Adrian, "Memory Management, the Win32 Subsystem, and Internal Synchronization in Chicago," *Microsoft Systems Journal*, May 1994, pp. 57-61.

Pietrek, Matt, "Stepping Up to 32 Bits: Chicago's Process, Thread, and Memory Management," *Microsoft Systems Journal*, August 1994, pp. 27-41.

Pietrek, Matt, "Investigating the Hybrid Windowing and Messaging Architecture of Chicago," *Microsoft Systems Journal*, September 1994, pp. 15-30.

Pietrek, Matt, "Which Win32 Is for You?," *PC Magazine*, September 27, 1994, pp. 303-308.

OS/2 Warp vs. Windows 95 – Multimedia		
Feature	**OS/2 Warp**	**Windows 95**
Image Viewer	Yes	No
Photo CD Support	Yes	No
Autodesk Animation	Yes	No
Play Any Audio File from Internet	Yes	No
Audio/Video Synch Manager	Yes	No
MPEG Support	Yes	Yes
32-bit Audio/Video Playback	Yes	Yes

OS/2 Warp vs. Windows 95 – Bundled Productivity Tools		
Feature	**OS/2 Warp**	**Windows 95**
Internet Access Tools	Yes	No
FTP	Yes	No
Telnet	Yes	No
Gopher	Yes	No
NewsReader	Yes	No
WebExplorer	Yes	No
CompuServe Front-End	Yes	No
Word Processor	Yes	No [12]
Spreadsheet	Yes	No
Database	Yes	No
Charting	Yes	No
Report Writer	Yes	No
Electronic Mail	Yes	Yes
Image Viewer	Yes	No
Fax	Yes	Yes
Phonebook	Yes	No
Personal Information Manager	Yes	No
System Information	Yes	No
Video IN	Yes	No
Video Conferencing	Yes	No
[12] Windows 95 comes with a simple text editor, not a word processor		

Development of OS/2 Warp Version 3

Dr. Khoa Huynh, Ron Cadima, and Jim Pascale
IBM Boca Raton Programming Center
Boca Raton, Florida

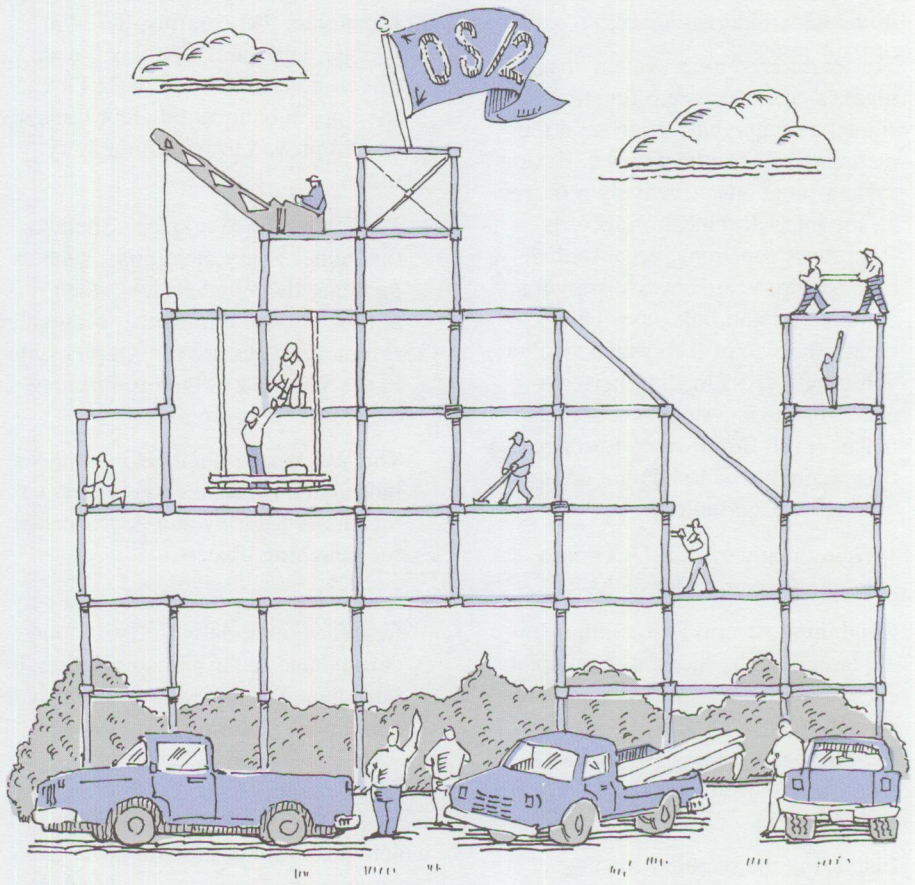
Reprinted with permission from Innovations, *the IBM Boca Raton site technical publication, Issue 3, 1994.*

The goals for OS/2 Warp Version 3, the latest version of Operating System/2 (OS/2), were relatively simple from a customer perspective, but technically challenging for the IBM Boca Raton Personal Operating Systems Programming Center development team.

Our goals were:

- OS/2 Warp Version 3 is to be an ideal desktop operating system that is robust, fast, easy to use, and capable of exploiting the latest software and hardware technologies.
- It must be a personally responsive OS/2, whose screen response time, command and window processing, and application performance must be noticeably improved for all current users of OS/2 for Windows.
- For new users of OS/2, the new OS/2 version must work well on their low-end computers with only 4 MB of memory.
- In addition, support for new software and hardware technologies, such as Plug and Play for PCMCIA services, object-oriented programming, and multimedia applications, must be incorporated into OS/2 Warp Version 3.

Needless to say, these goals were very ambitious, but through persistent and smart teamwork, our development team was able to deliver them all!



Four Major Areas

Work on OS/2 Warp, which began in earnest during the summer of 1993, can be divided into four areas:

- Optimized performance for desktop systems
- Improved application support
- Greater ease of use
- Wider hardware compatibility

These four areas are now discussed in more detail.

Optimized Desktop Performance:
One of the principal goals of OS/2 Warp was to work well in "constrained" environments; that is, on personal computers that have only 4 MB of physical memory. We have achieved this goal through performance enhancements and memory-usage reduction.

For performance enhancements, we examined all of our code base, and where possible, reduced the code paths in many areas of the system, including the kernel, Presentation Manager (PM), Workplace Shell, and DOS/Windows support code. Many design changes were also made to improve the overall system performance. Major design changes include:

- Providing the EXEPACK:2 option for executable, resource, and message files.

  This linker option reduces the size of files stored on disk, thereby reducing the number of I/O operations required to load them.

- Loading many OS/2 system functions into predetermined memory locations in the linear address space.

  This is referred to as system "basing," since each system function is assigned a unique base address in the memory linear address space. Basing system functions effectively removes all the internal relocation records from those functions, because their base addresses are already known at compilation and link time. This makes the size of the system smaller, and improves its loading performance. Basing a system function also makes it run faster, since fewer relocation records need to be applied in the event of a page fault.

- Merging many system DLLs into fewer and larger system DLLs.

- Combining several DLLs into a single large DLL reduces all redundant code and relocation records that are present in each DLL. It also makes our effort at reducing the memory working-set size of these DLLs much easier.

- Enhancing the memory-management subsystem to reduce the number of page tables required for shared code and data.

- Improving the memory-swapping algorithm so that the swap file is less fragmented, and the number of disk I/O operations that must be performed when the system runs out of memory is reduced.

  We also enhanced the swapping algorithm so that some memory pages, after being displaced from memory, are more optimally loaded back into memory from the less fragmented swap file than from their executable files.

- Enhancing the scheduling algorithms to improve task-switch times.

- Converting all important PM subsystems to 32-bit code, and implementing many algorithmic changes to boost windowing performance.

  In OS/2 Warp, the window management, the Workplace Shell, the graphics engine, PM controls, the print spooler, and the S3 display device driver are all rewritten in 32-bit C. Memory suballocation algorithms are also improved to reduce memory usage.

- Modifying the Workplace Shell so that unnecessary operations, such as updating the Window List when it is not visible, are eliminated. Many algorithmic changes are also implemented in the Workplace Shell to improve the folder bring-up time.

  The Workplace Shell also features a redesigned Find function, which has increased usability and performance for searching objects.

- Improving the print quality and performance, especially in both monochrome and color bitmap printing, by introducing new 32-bit printer drivers and a new 32-bit print subsystem.

For DOS applications, OS/2 Warp provides a VDM priority slider setting, which allows the prioritization of DOS sessions within their priority level. In other words, with this priority setting, the user now has more control over the amount of system resources dedicated to each DOS application.

For Windows applications, OS/2 Warp provides a Fast Load option. Under this option, a small Windows application is created to preload a Windows session into memory during system boot-up (this small application has no other function, and is not displayed in the Window List). Subsequent Windows applications can then be started in that same common session. Since these applications do not have to load the Windows environment into memory, their loading performance is much better. We have also provided an optimized DOS COMMAND.COM for use in the Windows environment.

In addition to the major design changes discussed above, we have also adjusted several system configuration parameters in the CONFIG.SYS file specifically for a system with only 4 MB of memory. During system boot-up, the system initialization program automatically adjusts the disk-cache size according to the amount of physical memory detected in the system. If the system has 4 MB of physical memory, the disk-cache size is reduced to 48 KB. The system installation program also adjusts the maximum number of threads to 96 during OS/2 system installation if the detected amount of memory is 4 MB. Since the operating system has to allocate memory for the disk cache and reserve memory for the maximum number of threads supported, reducing these configuration parameters increases the amount of memory available for other uses. These configuration parameter changes have proved to be very effective in many constrained-memory environments at 4 MB.

To further improve the overall system performance, we devoted much effort to reducing the system memory usage. This was critical in order to have a desktop operating system that works well in 4 MB of memory. Using a variety of internally developed tools, the memory-access patterns for both code and data were obtained for many important system components. These patterns were then used as a guide to rearrange code routines and data structures in order to minimize the number of memory pages required. This process is often called *page-tuning*. More specifically, functions, routines, and data structures that are frequently used together in space and time were placed and linked together in a way that minimizes the memory working set. In OS/2 Warp, the kernel and virtually all of the OS/2 system DLLs have been page-tuned.

Better Application Support: Win32s (versions 1.0 and 1.1) applications are now supported in OS/2. OS/2 Warp can be installed over Windows 3.11 and Windows for Workgroups (versions 3.1

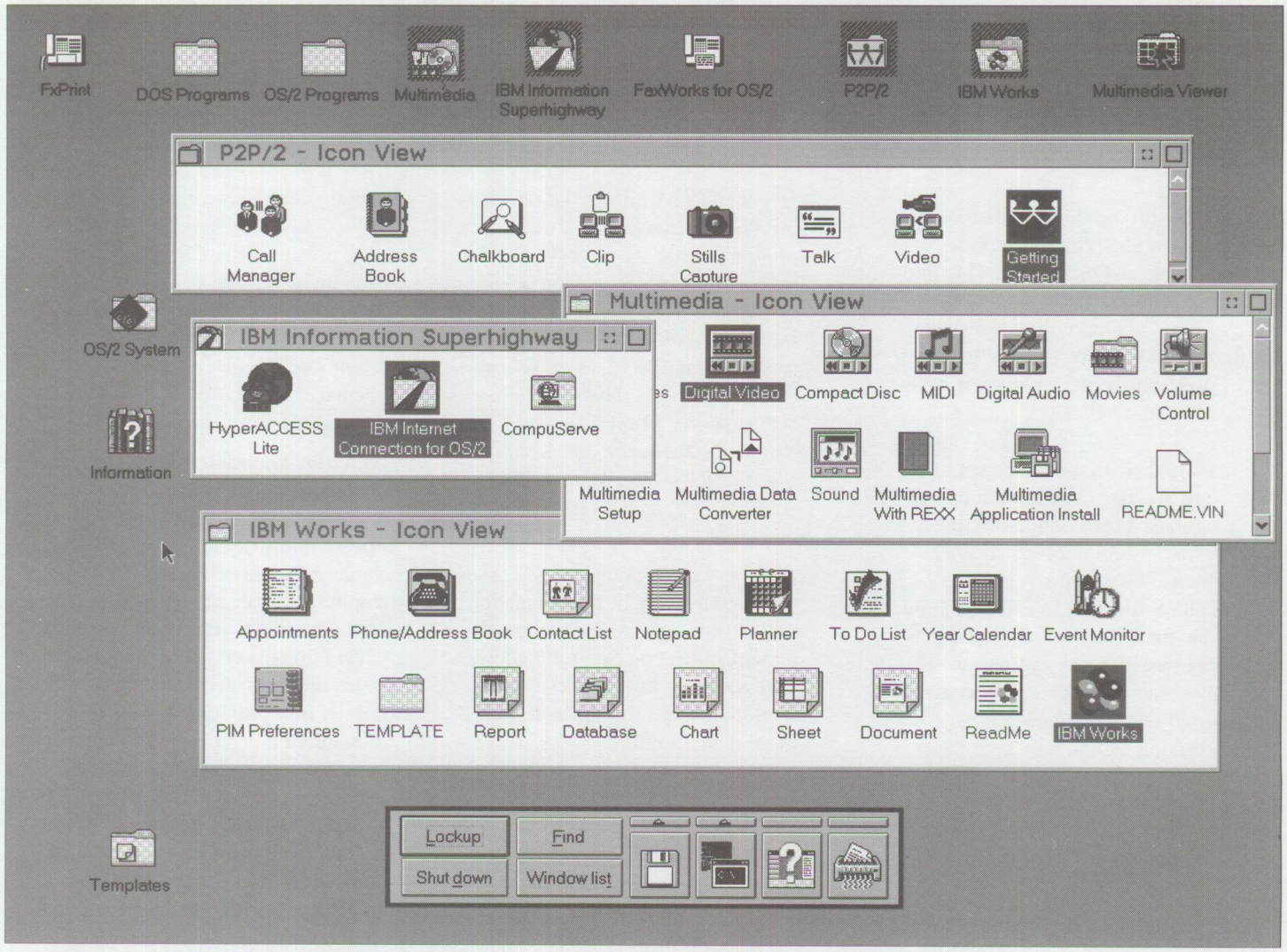


Figure 1. The OS/2 Warp Workplace Shell

and 3.11). However, the Windows networking functions are not supported under OS/2's WIN-OS/2. For object-oriented programming, support for the System Object Model (SOM) Version 2.0 and Distributed SOM (DSOM) has been added to to OS/2 Warp.

OS/2 Warp's multimedia enhancements make it a more exciting consumer-oriented system than ever before. It incorporates Kodak's Photo CD support, the industry's high-quality digital imaging standard. Support for the Motion Picture Experts Group (MPEG) file format, an emerging standard for digital motion video, is also provided in OS/2 Warp.

Improved Ease of Use: The usability of an operating system is very important if it is to find its way onto many desktop computers. The OS/2 development team is well aware of this, and has made significant changes in two major areas – the installation process and the Workplace Shell.

We redesigned the OS/2 installation program so that it can automatically configure the computer on which it is being installed when the user chooses the Easy Install option. This lifts a large burden from the user, since most users do not know off-hand all the needed details about the hardware components in their systems. Furthermore, the number of diskettes required to install OS/2 Warp is reduced, thanks to the new XDF installation diskette format. This new installation diskette format increases the amount of usable physical space on each diskette.

We also made numerous changes to the Workplace Shell for easier navigation. The Workplace Shell now features new graphics, including new icons, bitmaps, and color schemes (see Figure 1). It also introduces a customizable *LaunchPad*

for quick, convenient access to the user's most frequently used objects and programs. Additionally, the Workplace Shell allows users to save and restore their desktop configurations so that they can always return to their favorite setup.

Besides the work on the installation and the Workplace Shell, a new tutorial makes its debut in OS/2 Warp. This tutorial introduces users to the wonderful world of OS/2 at a pace they can select. The tutorial can also be tailored to each type of user and the user's level of experience with computers.

Also making their first appearance in OS/2 Warp are the configuration and recovery options. These options enable users, before system boot-up, to select previously used desktop setup files, CONFIG.SYS file, and INI files that are appropriate for the computer system currently being used. These options are especially convenient for laptop systems that are used with docking stations. Since a docking station has more hardware than the laptop itself, the user usually maintains separate sets of CONFIG.SYS, INI, and desktop setup files for the docking station and the laptop.

BonusPak

Today's desktop computer users need not feel isolated in their offices. Recognizing the trend toward the fully connected office, we have made available a BonusPak to accompany OS/2 Warp. We wanted to provide users with a set of productivity and online service applications that many desktop users would consider a true "bonus" package.

The useful applications in the BonusPak are:

- IBM Information Superhighway: an application package that includes the CompuServe Information Manager for OS/2, HyperACCESS Lite for OS/2, and Internet Connection for OS/2.
  - *CompuServe Information Manager for OS/2:* an application that gives users access to over 2,000 products and services by providing connection to many electronic and mail facilities available on CompuServe.
  - *HyperACCESS Lite for OS/2:* a 32-bit, object-oriented OS/2 modem program. "Lite" is an easy-to-use, reduced-feature-set version of HyperACCESS. The user can call bulletin-board systems (BBSs), Internet, CompuServe, or remote systems of all kinds. This program also supports 32-bit Zmodem, Ymodem, Xmodem, and Kermit file-transfer protocols, as well as ANSI, VT52, and VT100 terminal emulation.
  - *Internet Connection for OS/2:* a set of programs which, in conjunction with a service provider such as Internet Connection Services, allows user access to the Internet – a network that spans the globe and links more than 20 million users. The programs provided in this set can be used to send electronic mail (UltiMail), access online bulletin boards (NewsReader), explore the Internet (Gopher), access information on other computers (Telnet), connect to host computers which support 3270 sessions (3270 Telnet), or transfer files between remote computers (FTP).
- *Person-to-Person for OS/2* (P2P/2): a conferencing application that enables several users to work together without actually being together. The link between users can be made over modems or local-area networks (LANs). A set of utilities is provided to allow conference participants to carry out collaborative tasks. Among the most useful is the "chalkboard" utility, which provides a shared work area which all conference participants can see, annotate, and discuss the contents through a set of simple drawing and pointing tools. With appropriate ActionMedia II hardware, the "video" utility enables conference participants to originate and distribute video files, and to view video files created by other participants. This application requires at least 8 MB of memory.
- *IBM Works for OS/2:* a set of office automation application and productivity tools including Footprint Works, Document, Sheet, Chart, Database, Report, Template, PIM Preferences, Event Manager, Year Calendar, ToDo List, Planner, Notepad, Contact List, Phone/Address Book, and Appointments. This set of tools also requires at least 8 MB of memory.
- *Video IN for OS/2:* a set of applications and device drivers that allows the user to manipulate digital video files (commonly known as AVI files). Included in this package is a recorder application that can be used to create and edit digital video files in either IBM Ultimotion or Intel Indeo software motion video formats. An AVI File Utility provides more information about a given AVI file and permits the user to tailor various characteristics about the file, such as video/audio synchronization. Also included is a media player to control certain types of Pioneer laserdisc players.
- *IBM Multimedia Viewer:* a versatile tool for viewing and organizing multimedia objects, such as images, video, animation, audio, and text. This tool features a special type of folder, called the "Light Table folder," which has a number of special characteristics in addition to all the features of a standard OS/2 folder. Image objects within a Light Table folder appear as if they were photo slides on a photographer's light table. These slides are created when multimedia objects are moved or copied into a Light Table folder, thereby greatly simplifying the task of organizing and browsing multimedia data.
- *FaxWorks for OS/2:* a 32-bit software package fully enabled for the Workplace Shell. The user can send and re-

ceive faxes of unlimited length, and print them to any OS/2 printer.

- *IBM Installation Utility and System Information Tool:* a tool that enables the user to quickly and conveniently access detailed information about the hardware and software configuration of his or her computer.

Wider Hardware Compatibility:
Making its debut in OS/2 Warp, our new Plug and Play for PCMCIA interface has been designed to provide quick, convenient access to PCMCIA socket services for the following portable computer systems:

- Ambra 486 SO 425C
- AST Powerexec 4/25SL, 4/25SLC, and Bravo NB Color
- Compaq Concerto
- CompuAdd 425 TX
- IBM ThinkPad Models 350, 500, 720, 750, 755, and PS/2 E
- Panasonic CF-V11P
- NCR Safari 3180
- NEC Versa C, Versa E
- Toshiba Models T3600CT, T4700, T4800
- Zenith 425L
- ZEOS Colornote

Additionally, continuing our commitment to support as many hardware devices as possible in OS/2, we have incorporated support for new graphics accelerator cards, audio devices, and CD-ROMs. OS/2 Warp now supports the following new graphics accelerators:

- ATI Mach32/Mach64
- Cirrus 5428, 5430
- S3 864
- Tseng ET4000 W32p, W32i
- WD C24/C26/C27/C31/C33/C34
- Weitek P9000

Also included in OS/2 Warp is a set of popular OS/2 and Windows audio device drivers, including:

- Aztec Sound Galaxy
- Crystal Semi 4231
- ESS 688
- SoundBlaster, SoundBlaster PRO, PRO16, AWE32

Compared with OS/2 2.1, OS/2 Warp also supports the following additional CD-ROM devices:

- Philips Models 205, 205MS, 206, 225, 225MS, 226
- Sony Models 531, 535, 6150, 6251, 6201, 6205, 7201, 7205

OS/2 Warp Version 3 has all the "right stuff."

OS/2 Warp – The Right Stuff

With major improvements in performance (especially in low-end desktop systems with only 4 MB of memory), improved ease of use, Plug and Play for PCMCIA capabilities, wider hardware compatibility, and robust support for object-oriented programming and the latest Windows environments, OS/2 Warp should become entrenched as the 32-bit operating system of choice for many desktop users.

Built on the strengths of previous versions of OS/2 – the 32-bit programming model, priority preemptive multitasking, advanced Workplace Shell interface, object-oriented programming support, and seamlessly integrated support for DOS, Windows, and OS/2 applications on a single platform – OS/2 Warp Version 3 has all the "right stuff" to maintain its lead over other 32-bit operating systems.

Dr. Khoa Huynh *is currently on the technical staff of the OS/2 development manager. He has worked several years on performance issues for OS/2, proposing and implementing many system-design changes to improve the OS/2 system performance. He also developed several tools to optimize its performance. Dr. Khoa Huynh joined IBM in 1989. He is a patent holder, recipient of several IBM technical achievement awards, and has authored nearly 30 technical papers and patent applications. He is a member of the IEEE, and has chaired technical sessions at IEEE-sponsored conferences. His areas of interest include operating-system design and analysis, performance modeling, hardware system architectures, real-time computing, and multimedia systems. Dr. Khoa Huynh can be reached via Internet at khoa@vnet.ibm.com.*

Ron Cadima *is an advisory programmer in OS/2 Kernel Development within Personal Operating Systems, IBM Personal Software Products division, Boca Raton, Florida. He is responsible for scheduling, tasking, and semaphores, and for OS/2 system performance analysis. Ron has worked in OS/2 development for six years doing performance analysis and tuning, as well as kernel development. He has been with IBM for 27 years, primarily working on small systems and process control, developing operating systems for System/7, Series 1, point-of-sale systems, and OS/2. Ron's Internet userid is rcadima@vnet.ibm.com.*

Jim Pascale *is an advisory planner in OS/2 Planning, Personal Operating Systems, IBM Personal Software Products division, Boca Raton, Florida. He is currently the product planner for OS/2 Warp. For the past several years, Jim has worked in OS/2 development, and he was on the launch team for OS/2 2.0. He joined IBM in 1981.*

OS/2 2.x Service and Support: Installation (Part 2)

Kirk Krauss
Keane, Inc.
Boca Raton, Florida

This article is the second installment of the IBM publication OS/2 2.x and OEM Hardware, *produced by the Worldwide OEM Technical Service and Support group in Boca Raton, Florida. This book was issued by Robert J. Dilena and prepared by Kirk J. Krauss. The second installment resumes discussing OS/2 installation.*

During boot, from the time the system's power is turned on until the OS/2 Desktop appears, control of the system passes through several procedures that can be roughly described as follows:

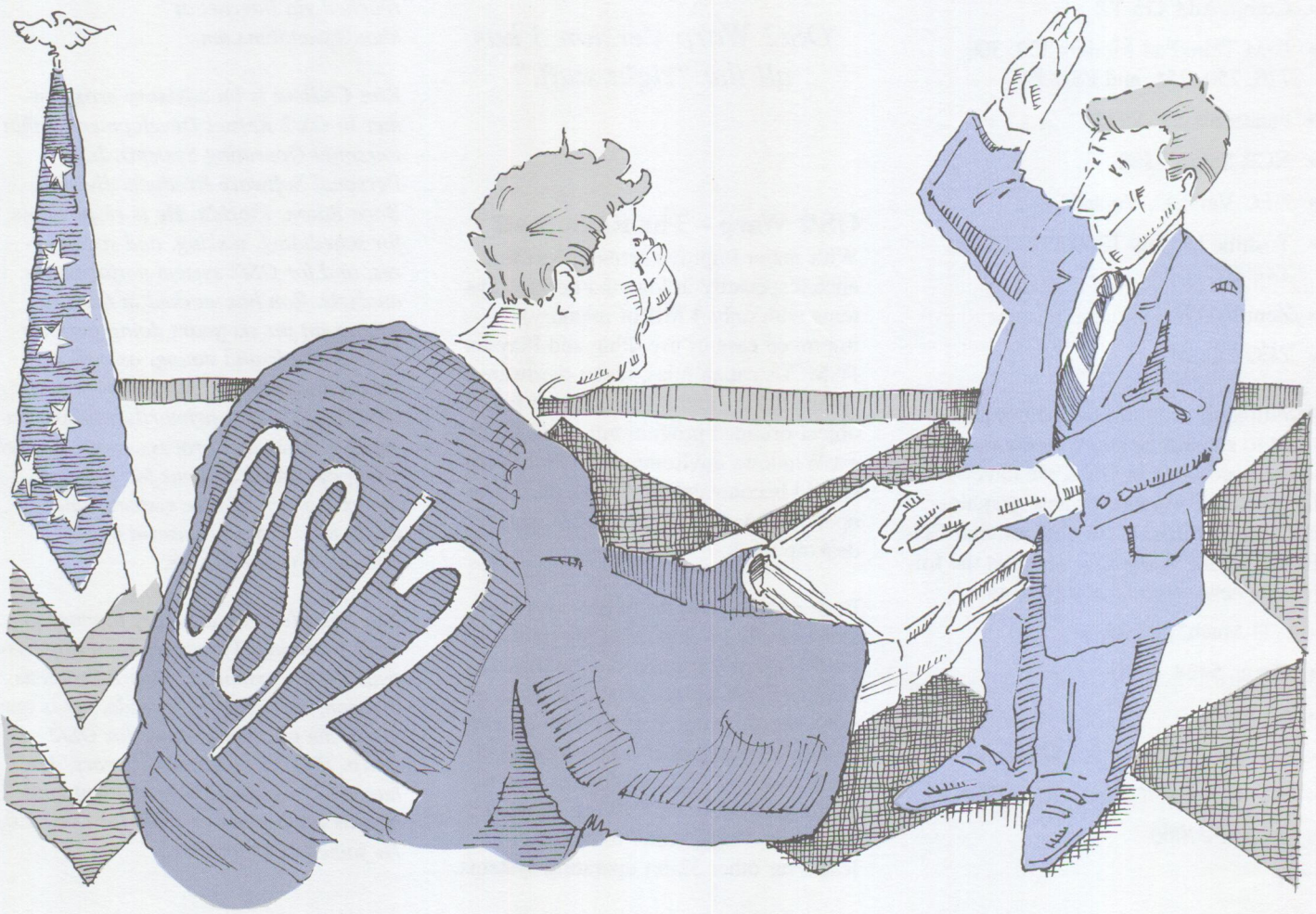
- Power-On Self Test (POST)
- Loading the OS2BOOT file
- Loading the OS2LDR file
- Loading the OS2KRNL file
- Processing the CONFIG.SYS file
- The Workplace Shell becomes operational

Each of these steps is described below.

POST

When the computer is switched on, a power-on self test begins. The power-supply voltage, the processor, and all the system's adapters are sequentially checked.

The system timer holds the processor's "reset" line ON until the power-supply voltage stabilizes. When the power becomes consistent, the processor begins functioning in real mode, with page swapping disabled, and it automatically executes the ROM code located at address FFFFFFF0h, which is 16 bytes below the top of the addressable memory space. This address contains a JMP instruction pointing to the ROM BIOS starting address.



The BIOS code may contain a test of the processor's functionality. The processor's self-test typically runs in less than .05 seconds. If this self-test does not pass, then the processor places a diagnostic value in the EAX register. Because the video adapter has not yet been initialized, any problems are reported to the user via audio signals on the system's speaker. The BIOS code then checks for the presence of a Floating Point Unit (FPU) or numeric coprocessor, which may be tested as well.

After the processor and FPU self-tests are complete, the BIOS instructions call for a search of the standard video ROM addresses, C0000h through C7800h. If video adapter ROM code is present, then BIOS subjects it to a checksum test. If no video adapter is found, then BIOS uses the motherboard's video ROM circuitry instead. In either case, the BIOS code calls the video ROM code, initializing the video system and placing a cursor on the screen.

With the video system operational, the search for adapter ROMs continues through the top of the standard adapter ROM address range, which may extend somewhat beyond E0000h depending on the type of BIOS used in the computer system. If any additional adapter ROMs are present, then the BIOS code performs a checksum test on each before executing it. Any checksum failure is reported as a "ROM Error."

BIOS determines whether the system was cold-booted or warm-booted by reading address 472h. A value of 1234h here indicates a warm boot; any other value indicates a cold boot. If the system was cold-booted, BIOS may conduct a simple RAM test.

When POST has completed without errors, the system beeps once. If there is a disk in the floppy drive, then BIOS loads and executes the code in its boot sector, which is the first sector on the disk. System files are then loaded from the disk. If there is no disk in the floppy-diskette drive, then BIOS loads and executes the first hard-disk drive's master boot record, which is the first sector on the drive. The last two "signature" bytes of the master boot record should be equal to 55AAh. Otherwise, a software INTerrupt 18 will occur, signalling the system to either run a ROM-based BASIC interpreter or display an error message.

The boot loader is specific to the partition in which it resides, and it cannot be successfully copied to another partition.

Multi-Boot Block: If Boot Manager has been installed, the active primary partition is the Multi-Boot Block (MBB). The MBB is the 1 MB partition that was set aside by FDISK when Boot Manager was installed.

The hard-disk drive's master boot record contains pointers to the MBB, and execution proceeds with the MBB's code. The code in the MBB determines which primary or logical partition to boot next. The display of the Boot Manager screen is actually one of several options available to the MBB, since the Boot Manager screen may be bypassed. Any partition (though not more than one) may be marked as "bootable"; the bootable partition is then booted by the MBB. Pointers to the partitions' boot blocks are stored as a linked list between the boot blocks themselves. In any case, the MBB loads and executes the boot loader for the bootable partition, as selected by the user.

The Partition Boot Record: The boot loader is found in the first sector (512 bytes) of any bootable partition. This sector is known as the *boot block* of the partition. The boot-loader code found here has normally been written by a utility such as SYS (for DOS) or SYSINSTX (for OS/2) to boot the operating system found in the partition. The boot loader is specific to the partition in which it resides, and it cannot be successfully copied to another partition. Of course, the boot loader is also quite specific to the operating system that it boots.

The Super Boot Block: Under the HPFS file system, the one-sector boot block is contiguous with a 16-sector *super boot block.* Unlike the boot block, the super boot block is not partition-specific, and in fact the super boot block may be copied between HPFS partitions if boot problems arise.

The super boot block contains a real-mode file system driver, known as the micro-FSD, which is specific to HPFS. FAT partitions contain no super boot block.

The OS2BOOT File: Under the OS/2 2.x version of the FAT file system, known as Super-FAT, the OS/2 partition's boot loader loads and executes OS2BOOT, which is the operating system boot loader for OS/2. FAT support is included in both the kernel loader and in the OS/2 kernel itself. Under HPFS, the super boot block is loaded and executed before control is transferred to OS2BOOT. OS2BOOT loads a protect-mode file system drive called the mini-FSD.

The OS2BOOT code finds and loads OS2LDR, the kernel loader. The sectors comprising OS2LDR may be fragmented across discontiguous clusters – hence the need for OS2BOOT. OS2LDR loads under a basic INT 13 device driver.

The OS2LDR File: The kernel loader, OS2LDR, initializes the Global Descrip-

tor Table, Local Descriptor Table, and Interrupt Descriptor Table registers in the processor, so that the OS/2 kernel can switch the processor into protect mode. The kernel loader utilizes the micro-FSD to load the OS/2 kernel's various object modules into RAM, then it executes the kernel system's initialization component.

The OS2KRNL File: The OS/2 kernel, OS2KRNL, loads the base device drivers specified in the CONFIG.SYS file, then it switches the processor into protect mode so that it can start using these drivers. The kernel continues processing the CONFIG.SYS file, loading device drivers and other software components. The kernel then passes file-system control from INT 13 to the file system's device manager (the .DMD file). The boot partition's own system-specific FSD, determined by the `IFS=` statement in the CONFIG.SYS file, is then loaded and initialized. Finally, if the `PROTSHELL=` statement calls PMSHELL.EXE, then the Presentation Manager Workplace Shell routines, found in the PMWP.DLL file, and the graphics engine routines, found in the PMGRE.DLL file, are invoked.

The CONFIG.SYS File

You can modify many aspects of your OS/2 configuration by making changes to the CONFIG.SYS file, which appears in the root directory of the hard drive where OS/2 is installed. (A backup of the original CONFIG.SYS file is also stored in the \OS2\INSTALL subdirectory when OS/2 is installed.) The CONFIG.SYS file determines the file system and device drivers to be used, the path and other environment variables, tuneable performance-related settings, and other system information.

You can modify many aspects of your OS/2 configuration by making changes to the CONFIG.SYS file.

The CONFIG.SYS is not processed in the order of its statements. Instead, the OS/2 kernel reads this file in several passes, searching for different types of directives each time. Duplicate statements can create boot problems and should be avoided. Directives are processed in the following sequence:

```
SET
DISKCACHE=
BASEDEV=*.SYS
BASEDEV=*.ADD
BASEDEV=*.I13
BASEDEV=*.FLT
BASEDEV=*.DMD
DEVICE=    (Physical Device Drivers only)
IFS=
RUN=
CALL=
PROTSHELL=
DEVICE=    (Virtual Device Drivers)
```

The CONFIG.SYS statements and associated parameters described in Figure 1 (below and on the following pages) do not comprise a complete reference; however, descriptions of statements commonly altered by users of OEM systems and by technical support representatives are included.

Commonly Altered CONFIG.SYS Statements

`IFS=[drive] [path] filename [arguments]`
Installs an Installable File System (IFS) by specifying the path and file name of the file system driver. The file system driver contains code needed to manage drives and floppy disks formatted for any file systems other than FAT. This line is required for systems that use the HPFS file system.

The `/C:nn` argument is used to set up drive caching under HPFS. The `nn` parameter sets the size of the cache (in KB).

`PROTSHELL=[drive] [path] filename [arguments]`
Loads the OS/2 command processor. Normally, `PROTSHELL=` should be set to `PMSHELL.EXE` to start the Workplace Shell. If this line is not present, the default OS/2 user interface, CMD.EXE, is loaded.

`SET string= [string]`
Sets one string in the environment equal to another string for use as search paths and other environment variables. The `SET` command can be used in the CONFIG.SYS file, as well as in .BAT and .CMD files.

Figure 1. (1 of 6) Commonly Altered CONFIG.SYS Statements

Commonly Used OS/2 Environment Variables

SET AUTOSTART=[PROGRAMS] [,TASKLIST] [,FOLDERS] [,CONNECTIONS]
This variable is primarily used to automatically restart processes that were running when the system was last shut down. **PROGRAMS** causes applications to be automatically restarted. **FOLDERS** reopens Desktop folders. **CONNECTIONS** re-establishes network connections. **TASKLIST** enables the Window List Desktop object, which displays a list of running jobs.

SET RESTARTOBJECTS=YES | NO | STARTUPFOLDERSONLY [,REBOOTONLY]
This statement determines which processes are started by **AUTOSTART**. The default is **YES.** If **NO** is chosen instead, only the desktop starts when OS/2 is booted. The **STARTUPFOLDERSONLY** parameter causes only applications in the Startup folder to be started. When the **REBOOTONLY** parameter is added, processes are started when the system is rebooted, but not when the Desktop is restarted after it stopped, which usually happens only after a Workplace Shell error has occurred.

SET USER_INI=[drive] [path] filename
SET SYSTEM_INI=[drive] [path] filename
These two variables are normally set to **OS2.INI** and **OS2SYS.INI,** respectively. The .INI files include information about the Workplace Shell setup and the hardware configuration. You can copy your OS2.INI file to a file of a different name, reboot the system, and then change your OS/2 Desktop. Subsequently, you can choose between two Desktop layouts by changing the **USER_INI** variable setting in the CONFIG.SYS file, and then rebooting.

SET OS2_SHELL=[drive] [path] filename
The Workplace Shell executes this file when you select OS/2 Full Screen or OS/2 Window. Normally, **OS2_SHELL** is set to **CMD.EXE,** the OS/2 command interpreter.

SET RUNWORKPLACE=[drive] [path] filename
This variable should be set to **PMSHELL.EXE** in order to activate the Workplace Shell.

SET COMSPEC=[drive] [path] filename
This variable should be set the same as **OS2_SHELL.** Some functions use this variable to load parts of the OS/2 command processor, normally CMD.EXE.

SET PATH=[drive] path [;[drive] path]
SET DPATH=[drive] path [;[drive] path]
SET HELP=[drive] path [;[drive] path]
SET GLOSSARY=[drive] path [;[drive] path]
These search paths are used for executable program (.EXE) files, program data files, and help and glossary (.HLP) files, respectively.

SET PROMPT=
You can define a different command prompt for the OS/2 command interpreter by using this variable. Since ANSI.SYS support is built into OS/2, you can set your prompt to a different color. For example, the command

```
SET PROMPT=$E[33;1m[$p]$e[34;44m$e[36;1m
```

changes the screen colors to light-blue on dark-blue, and the prompt will be the working directory, shown in yellow.

SET DIRCMD=
If this variable is set to **/O:GN,** the DIR command will display directories in alphabetical order.

Figure 1. (2 of 6) Commonly Altered CONFIG.SYS Statements

LIBPATH=[drive] path [;[drive] path]
Identifies a set of directories to be searched when the system loads Dynamic Link Libraries (DLLs). The current directory is not searched first. To have the current directory searched first, add its path

```
.;
```

to the beginning of the **LIBPATH=** statement.

PRIORITY_DISK_IO=YES | NO
When this parameter is set to **YES**, an application running in the foreground receives disk I/O priority over applications running in the background. Use this to improve the response time of foreground applications.

TIMESLICE=min [,max]
Determines the minimum and maximum amounts of time that a thread may be active before the system switches to another task.

FILES=n
Determines the maximum number of files available in VDMs. When a VDM is started, 20 files are available to be used by all programs, but an application may increase the number of files up to the limit defined by the **FILES=** statement.

DEVICE=[drive] [path] filename [arguments]
Installs a device driver, which is a program permitting OS/2 to recognize a device and to process data passed to and from that device. **DEVICE=** statements are processed in the order in which they appear in CONFIG.SYS.

The standard OS/2 device drivers are:

ANSI.SYS	Provides extended screen/keyboard support for DOS sessions
COM.SYS	Allows applications to use serial devices
EGA.SYS	Provides EGA support for DOS sessions
EXTDSKDD.SYS	Provides an additional logical drive letter to an existing floppy-disk drive
LOG.SYS	Provides system error logging using the SYSLOG utility
MOUSE.SYS	Provides pointing-device support
PMDD.SYS	Provides pointer draw support for OS/2 sessions
POINTDD.SYS	Provides mouse pointer draw support
TOUCH.SYS	Provides support for touch devices
VDISK.SYS	Installs a simulated disk for DOS sessions
VEMM.SYS	Supports the DOS Expanded Memory Manager
VXMS.SYS	Supports the DOS Extended Memory Specification

Printer drivers are provided on the Printer Driver diskettes in the OS/2 installation set.

An additional driver called **FSFILTER.SYS** is used for Virtual Machine Boot (VMB) environments, allowing some DOS versions to access OS/2 files.

BUFFERS=x
Sets the number of disk buffers that the system can use. These buffers are each 512 bytes in size, and they are used to transfer blocks of data that do not occupy complete sectors. Increase this number for systems that are expected to run numerous applications at once. Set it lower if the amount of RAM is low (approximately 4 MB).

IOPL=NO | YES [|proc2[,proc3]]
Allows I/O privileges to be granted to processes. Determines which code segments and data segments a process can access, and which instructions it can execute. The default privilege level is 3.

Figure 1. (3 of 6) Commonly Altered CONFIG.SYS Statements

DISKCACHE=n[,LW] [,T] [,AC:x]
Defines the number of blocks of storage, in KB, allocated to the hard-drive cache under the FAT file system. This statement does not apply to HPFS. The parameter **n** specifies the size of the cache in KB; **LW** enables lazy writes; **T** sets the cache threshold (this should normally be set to 32); and **AC:x** enables AutoCheck for a specific drive **x.**

Guidelines for specifying the parameter **n** are: For 5 MB or less of RAM, use a 64 KB cache; for 6 MB or more of RAM, use a 256 KB cache.

MAXWAIT=x
Sets the maximum time, in seconds, that a process will wait before its priority is increased.

PRIORITY=ABSOLUTE | DYNAMIC
Normally, this statement is not present in the CONFIG.SYS file. When the statement is not present, OS/2 varies the priority of each thread, depending on such factors as its age and whether it is running in the foreground. If this statement is present and is set to **ABSOLUTE**, the system will not change thread priority over time.

MEMMAN=[SWAP | NOSWAP] [,MOVE | NOMOVE] [,PROTECT]
Selects storage allocation options. If there is not enough memory to load a required page, the least frequently used data pages will be stored in the SWAPPER.DAT file on the hard-disk drive. Unused code pages may be discarded completely, since they can be reloaded from the original executable files if needed.

When the system is booted from floppies, swapping is not possible, and the default parameter is **NOSWAP. NOSWAP** is also a good choice for systems that run time-dependent applications. Use **PROTECT** to allow APIs to use protected memory. **MOVE** is a holdover from OS/2 1.x; it permits defragmentation of segments in RAM. Sixteen-bit OS/2 applications still use the segmented memory model under OS/2 2.x.

SWAPPATH=drive [path] [minfree] [initial]
Specifies the size and location of the swap file, SWAPPER.DAT. The **minfree** and **initial** values represent the free space that must remain in the drive partition, and the initial size of the SWAPPER.DAT file, respectively. If the SWAPPER.DAT file grows until the free drive space is less than the **minfree** value, a warning message is displayed.

BREAK=ON | OFF
Enables checking for Ctrl-Break. OS/2 intercepts this key sequence in any case, but it does so more quickly if you set **BREAK=ON.** Performance may be improved if you set **BREAK=OFF.**

THREADS=x
Specifies the number (between 32 and 4095) of threads that may execute at once. If the amount of RAM is low (4 MB), reduce this value from 256 (the default) to 128.

PRINTMONBUFSIZE=x,x,x
Sets the parallel port driver's character monitor buffer size. The arguments correspond to the sizes, in bytes, of the buffers for LPT1, LPT2, and LPT3, respectively. Increasing these values may improve the performance of parallel devices. The maximum buffer size is 2048 bytes.

COUNTRY=nnn [,[drive] [path] filename]
Identifies the following information for a country:

- Date and time format
- Decimal separator
- Character-case map table
- Collating-sequence table used by SORT
- Double-byte character set (DBCS)

The **filename** parameter specifies the file containing the country information.

Figure 1. (4 of 6) Commonly Altered CONFIG.SYS Statements

SET KEYS=ON | OFF
Enables the "retrieve" key (up arrow) for the OS/2 command interpreter when set to **ON**.

REM SET DELDIR=drive DELETE, n[;drive DELETE, n]
The REM preceding a command renders the command inoperative. Un-REM this statement to enable the UNDELETE command for the partitions specified in the drive arguments. Use of UNDELETE may consume a significant amount of drive space in the specified partitions, particularly if **n**, the number of dedicated files to be stored, is set to a high value.

BASEDEV=filename [arguments]
Installs a base device driver. The driver must be in either the root directory or the \OS2 subdirectory. Files are processed in order of their filename extensions.

The OS/2 standard base device drivers are:

PRINT01.SYS	for printers on non-Micro Channel* computers
PRINT02.SYS	for printers on Micro Channel computers
IBM1FLPY.ADD	for disk drives on non-Micro Channel computers
IBM2FLPY.ADD	for disk drives on Micro Channel computers
IBM1S506.ADD	for non-SCSI hard drives on non-Micro Channel computers
IBM2ADSK.ADD	for non-SCSI hard drives on Micro Channel computers
IBM2SCSI.ADD	for Micro Channel SCSI adapters
IBMINT13.I13	general-purpose hard-drive support for non-Micro Channel computers
OS2DASD.CMD	general drive support
OS2SCSI.DMD	general support for non-disk SCSI devices

PROTECTONLY=NO | YES
Enables both DOS and OS/2 environments when set to **NO**. Changing the default **NO** parameter to **YES** allows OS/2 sessions to use RAM below 640 KB, which is otherwise reserved for DOS sessions.

SHELL=[drive] [path] filename [arguments]
Loads and starts COMMAND.COM, the DOS command processor, or allows you to replace it with another command processor. The arguments are specific to the command processor.

FCBS=m,n
Defines File Control Block (FCB) information for DOS sessions. An FCB contains all the information about a file (structure, length, name, ...)

RMSIZE=x
Specifies the highest storage location allowed for DOS sessions. This can be used to limit the size of the DOS environment.

DOS=[HIGH | LOW] [,UMB | NOUMB]
Specifies whether the DOS kernel will reside in the high-memory area, and whether DOS applications will control Upper Memory Blocks (UMBs).

DEVINFO=
CODEPAGE=xxx [,yyy]
These two statements prepare a device for code-page switching, and selects the system codepages (defined character sets) to be prepared. Appropriate **DEVINFO** statements must be included in CONFIG.SYS so that the associated **CODEPAGE** statements can be of use. The format of the **DEVINFO** line depends on the type of device (keyboard, display, or printer) to which it is applied.

Figure 1. (5 of 6) Commonly Altered CONFIG.SYS Statements

RUN=[drive] [path] filename [arguments]
Starts a system program during system initialization. Workplace Shell programs cannot be started at this time.

CALL=[drive] [path] filename [arguments]
The CONFIG.SYS file can include a command such as

```
CALL=C:\OS2\XCOPY.EXE C:filename D:filename
```

You can protect your .INI files by backing them up each time OS/2 is booted. The following two lines back up the current .INI files as well as the last backup (assuming OS/2 is installed on drive C):

```
CALL=C:\OS2\XCOPY.EXE C:\OS2\*.INX C:\OS2\*.INY
CALL=C:\OS2\XCOPY.EXE C:\OS2\*.INI C:\OS2\*.INX
```

TRACE=ON | OFF [x [,x]]
TRACEBUF=x
OS/2 comes with a system trace facility that can be switched on to log system events and function calls. The TRACEFMT.EXE utility is used to analyze the trace output file. Each type of event and function has an associated number in the range of 0 to 255; such numbers can be specified as parameters in the **TRACE=** statement. The **TRACEBUF=** statement defines the size of the buffer in KB; the valid range is 1 to 64, and the default value is 4.

Figure 1. (6 of 6) Commonly Altered CONFIG.SYS Statements

Boot Error Messages

During the OS/2 boot process, enigmatic error messages such as **OS/2!!SYS2025** may appear when the system has a problem. If OS/2 is up and running, and a SYS... message is displayed on the screen, you may often find out more about its meaning by typing **HELP SYSxxxx**, where **xxxx** is the number displayed. The help provided in this way is generally quite good, and it often contains advice about how to remedy the error condition. But the error codes that can appear when the system is booting do not have associated help pages.

Figure 2 lists common messages that may appear at boot time.

The OS/2 Environment

The Workplace Shell is not invoked until the CONFIG.SYS file has been almost completely processed; all OS/2 environment variables have been set; all physical device drivers and device managers have been loaded; and any required installable file systems have been installed.

Some applications or utilities may start automatically at this time, unless the Ctrl-Shift keys are pressed or the **AUTOSTART** environment variable (often set in the CONFIG.SYS file) is altered.

In the Workplace Shell, icons are used to represent objects such as applications, printers, drives, functions, and folders. Icons can be arranged, moved, created, and deleted. The user may move icons

OS2!!SYS1475	The file OS2BOOT cannot be found. It may also be the case that the OS2BOOT file cannot be opened, or perhaps a non-bootable diskette formatted under OS/2 2.x is in the floppy drive.
OS/2!!SYS2025	A disk read error has occured. OS2BOOT may be corrupted and unreadable, or a non-bootable diskette may be in the floppy drive.
OS/2!!SYS2026	The file OS2LDR cannot be found. The OS2LDR file is either not present on the drive or it cannot be opened. Possibly there is a non-bootable diskette formatted under OS/2 1.x in the floppy drive.
OS/2!!SYS2027	Insert a system disk and restart the system. This message usually accompanies one of the other messages when the system stops.
OS/2!!SYS2028	The file OS2KRNL cannot be found.
OS/2!!SYS2029	The file OS2KRNL is not acceptable. It may be corrupted.
OS/2!!SYS2030	The system does not have enough storage to start the operating system. Add RAM or modify the CONFIG.SYS file.
OS/2!!SYS3146	The system cannot find the OS2LDR.MSG file. Ensure that this file is present on the drive.
OS/2!!SYS3147	The OS2LDR.MSG file is not valid. It should be replaced by a new copy.
OS/2!!SYS3161	A 386SX or higher processor is required. OS/2 2.x does not install on systems equipped with 80286 or lower processors.

Figure 2. OS/2 2.x Boot Error Messages

into a folder by dragging them across the Desktop with a mouse.

Some key sequences are useful for working quickly with the Workplace Shell and for navigating OS/2 without a mouse if one is not available or not working. The most useful key sequence is Ctrl+Esc, which brings up the Window List. From there, you can select any of the programs or windows that are open by selecting one with the up and down arrow keys and pressing Enter.

Other useful key sequences are listed in Figure 3.

The STARTUP.CMD File: There is a Startup Desktop folder as well as a STARTUP.CMD file. These are separate mechanisms for starting processes automatically when OS/2 is booted. Commands placed in the STARTUP.CMD file will process at startup in the order in which they appear. If an executable program's icon is placed in the Startup folder, that program will also start automatically.

The System Setup Folder and Selective Install: The System Setup folder is found inside the OS/2 System folder. In OS/2 2.1, this folder is available as a selection in the Desktop's pop-up menu. Some contents of the System Setup folder are shown in Figure 4.

In addition to the objects listed in Figure 4, the System Setup folder also contains Mouse, Keyboard, Font Palette, Sound, and System Clock objects. On SVGA systems, the System Setup folder also contains a Display Driver Install object.

Selective Install utilizes the INSTALL.EXE routine to obtain files from the OS/2 installation diskettes or CD-ROM. If a music CD is in the CD-ROM drive when the user starts Selective Install, then a SYS!!0045 error may result.

The Desktop Settings: With the mouse pointer on a clear portion of the Desk-

Key Sequence	Function
F1	Help
F2	General help
F9	Keys help
F11	Help index
Esc	Switch to previous Help screen
Space	Pick and unpick objects
Ctrl+/	Pick all objects in a window
Ctrl+\	Unpick all objects
Shift+F8, cursor keys, and Space, then Shift+F8 when done	Pick multiple objects
Enter	Start/select objects or answers
Shift+F10 after selecting the object(s)	Delete, print, move, or copy objects
Ctrl+Esc, choose "Desktop - Icon View," then Ctrl+\ (to deselect all objects), then Shift+F10	Open the Desktop's menu (permits shutdown without a mouse)
Alt+Tab	Switch between windows
Alt+Esc	Switch between both windows and fullscreen sessions
Alt+Space or F10	Show a window's menu
Alt	Close a menu
Alt+Space, Open Settings	Show settings
Alt+Page Up, Alt+Page Down	Page through settings
Alt+F4	Close a window
Alt+F6	Switch between a window and its help window
Alt+F7 or Alt+M, cursor keys	Move a window
Alt+F8 or Alt+S, cursor keys	Resize a window
Alt+F9 or Alt+N	Minimize a window
Alt+F10 or Alt+X	Maximize a window
Alt+F5 or Alt+R	Restore a maximized window
Alt+F11 or Alt+H	Hide a window
Page Up, Page Down	Page through text in a window
Shift+Delete	Cut text to the clipboard
Ctrl+Insert	Copy text to the clipboard
Shift+Insert	Paste text
Alt+Home	Switch between DOS window and DOS full screen
Print Screen	Print text window
Shift+Print Screen	Print entire Desktop
Ctrl+Alt+Print Screen	Force application to release the parallel port
Ctrl+Alt+Num Lock+Num Lock	Start stand-alone dump process

Figure 3. Useful Key Combinations in OS/2 2.x

top, you can tap the right mouse button, then select Open, then Settings, to get to the Desktop settings. Here, you can specify the manner in which Desktop icons are displayed, minimized, and hidden. You can also set a lockup password, and select background and lockup screen bitmaps.

The DOS Settings: There are two sets of DOS settings. You can find the general DOS settings by bringing up the OS/2 System folder and pressing the right mouse button on the DOS Window icon. You can also change the settings of an open DOS window by selecting the top left box by the window's label and choosing DOS Settings from the listbox provided.

The DOS/Windows Boot Sequence: A review of the boot process for DOS and Windows, compared to the OS/2 boot process, may be helpful at this point.

Following POST, if there is no diskette in the floppy drive, then the hard-disk drive's master boot record is loaded and executed. The master boot record's program searches the drive for logical partitions. A valid DOS volume boot sector should be present in the bootable partition.

The DOS volume boot sector is loaded, and it in turn loads IBMBIO.COM and IBMDOS.COM – if the partition was prepared using the FORMAT and SYS commands in IBM PC DOS. If MS-DOS was used, the DOS volume boot sector loads IO.SYS and MSDOS.SYS. If one of these sets of hidden files does not appear in the first two files in the directory, then an error message is displayed.

The DOS volume boot sector passes control to IBMBIO.COM (in IBM PC DOS) or IO.SYS (in MS-DOS). This code copies itself into the top of contiguous DOS RAM, resumes execution from there, performs some housekeeping, and then executes IBMDOS.COM (in IBM PC DOS) or MSDOS.SYS (in MS-DOS). This DOS kernel then loads the base device drivers, resets any connected devices, loads the file system, and returns control to the I/O system (IBMBIO.COM or IO.SYS).

Object	Function
Selective Install	Installs features that were not selected during the initial installation
Color Palette	Sets background colors
Scheme Palette	Sets colors for dialog boxes, text, radio buttons, etc.
Migrate Applications	Create icons for applications found on the system; use PARSEDB to modify the migration database
Spooler	Disables the print spooler or relocates it to a different partition
System	Changes global Workplace Shell settings
Device Driver Install	For new device drivers
Country	Updates the country settings

Figure 4. Contents of OS/2 2.x System Setup Folder

The CONFIG.SYS file is read four times, first by the DOS kernel and then by the I/O system. Statements are processed in a standard order, and base device drivers are loaded on the first pass. The second pass loads `DEVICE=` statements. The third pass handles `INSTALL=` statements. On the fourth pass, the `SHELL=` statement is processed. This statement loads a command processor. If no `SHELL=` statement is present, then the COMMAND.COM command interpreter is loaded by default.

COMMAND.COM loads and executes AUTOEXEC.BAT. If AUTOEXEC.BAT does not run any applications, then a DOS command prompt appears. If AUTOEXEC.BAT is not present, then COMMAND.COM executes the `DATE` and `TIME` commands, shows its copyright date, and then provides a DOS prompt.

If Windows has been installed, you can enter `WIN` to invoke the Windows shell. To diagnose Windows boot errors, you may enter `WIN /B` to create a log file named BOOTLOG.TXT.

Kirk Krauss *is a Consultant for Keane, Inc. in Boca Raton, Florida, working within Worldwide OEM Technical Services and Support in the IBM Personal Software Products division. He supports OS/2 Warp, performing defect analysis in a test lab made up of OEM systems. Previously, Kirk supported mainframe operating systems, and has done development work in UNIX, X-Windows, and OS-9. He has a BS degree in electrical engineering and a second BS degree in computer engineering, both from North Carolina State University. Kirk is a keyboardist who plays and composes music. His Internet userid is kirkk@bcrvm1.vnet.ibm.com.*

The OS/2 Corner: A Personal Introduction to OS/2

Len Zakas
Channel Islands PC Users' Group
Oxnard, California

Reprinted with permission from The Outer Edge, *newsletter of the Channel Islands PC Users' Group, October and November 1993 issues, and subsequently updated by the author for publication in this magazine.*

DOS is great and very capable. With some very simple menu programs, DOS can be made very easy for anyone to run any DOS-based program of choice. In fact, IBM's and Microsoft's recent upgrades to DOS are very significant for DOS-only users.

I see Windows 3.x as a glorified menu program that requires DOS to run. It is not an operating system, but an operating environment. Windows claims to provide unique and absolutely necessary capabilities for the personal home/desktop computer. Many Windows 3.x applications are indeed outstanding.

But trying to use those Windows capabilities requires you to know a lot about the basic operating system, DOS; a lot about how a computer works; and a lot about how Windows accomplishes what it does.

Taking the Time to Learn OS/2

Learning how to use OS/2 is only marginally harder (instead of a lot harder) than learning to use DOS or Windows.

The months-long time frame it takes to get comfortable with DOS and Windows is the same amount of time you should allocate for OS/2. In fact, here are some interesting questions for Windows users:

- How many months have you had Windows?
- What percentage of its capabilities do you know how to use?
- How many have you actually used?
- Why is there a continuing need for Windows Special Interest Groups (SIGs) if Windows is so easy to learn and operate?

To help me learn OS/2, I've depended on my user group, the Channel Islands PC Users' Group, for a lot of questions and answers – especially the answers. My user group has established an OS/2 SIG, and two other OS/2 user groups are within a 40-minute drive.

To any member of a local user group, depending on IBM is, at best, a second choice. Happily, I have found IBM to be very supportive and successful in solving my problems. Their 800 number has gotten me results.

Why I Switched to OS/2

I liked DOS and, with one exception, DOS did everything I wanted to do. But I have a statistics program that requires Windows 3.1. So, if I had to leave DOS, why not take the time to learn to use an operating system that takes advantage of the 80486 chip in my computer?

OS/2 has many more possible option settings for each program than are offered by DOS or Windows/DOS. There are about ten possibilities, plus some alternatives to even these.

For example, with three clicks of a mouse button, OS/2 gives you a standard DOS prompt, or a screen called WIN-OS/2 that looks a lot like the Windows 3.1 Program Manager screen. Or, you can select any combination or multiples of screen windows, and/or select an OS/2 prompt. Or, you can click on a screen icon to immediately start any DOS or Windows program, without bothering with a C: prompt or the Windows Program Manager. And all of the selected programs can be running at the same time!

The OS/2 for Windows product offers even the casual user an advanced operating system in an exciting package.

Yea for OS/2 2.1 Performance

When my DOS data manager program automatically updates a block of numbers under DOS or Windows, it allows me to walk from the back room to the kitchen, pour a cup of coffee, stir in the cream, and come back to the computer to watch it finish the calculations.

When I do the same updating using the Data Manager program icon on the OS/2 Desktop, it is half finished by the time I get to the door of my room (bad news if I really wanted that cup of coffee!).

This speed improvement is found in all my DOS programs. My Windows programs speed up a bit, but not as dramatically.

Under Windows, my Data Manager program had a problem finding its data files. Also, under DOS and Windows, this program and several other DOS programs had a timing problem with my computer, which required me to strike an action key more than once (as in "Press Y to continue"). OS/2 2.1 has fixed both of these glitches. The reason seems to lie with OS/2's taking control of all the computer's hardware. If there are any hardware problems that you may have not even noticed yet, OS/2 will find them!

Program Isolation

My simplified way of understanding what OS/2 does is to first picture how DOS runs a program. DOS loads the program and data into the first 640 KB of memory. Then, as the program runs, DOS prioritizes use of the CPU chip. It also manages the program's require-

ments for the monitor, disk space or data from the disk, and additional memory if the program can use it. Because DOS operates this way, and because Windows is only a menuing program, each and every individual Windows program must be correctly written to allow DOS to control these resources properly while allowing DOS to perform in the background. If a Windows program is not written properly to operate with DOS, or to properly communicate priorities with other programs, the entire computer freezes up.

OS/2 does a similar management job, except that it looks at all the available memory as a big block. For each program loaded, it will place the program in whichever block of memory it chooses. When a second program is loaded, OS/2 isolates this program from all others, and then manages the computer resources (the 386/486 chip itself, the monitor, sound cards, and so on) among all the active and inactive (background) programs. The OS/2 operating system, not each individual program, controls all the programs and the multitasking functions between them.

This isolation of programs allows for optimizing each program that you choose to run under OS/2. In addition to the standard 60-lines-plus CONFIG.SYS file in OS/2, each program has over 60 additional possible settings, an ability to call for unique drivers, and the option of creating a unique AUTOEXEC.BAT file for each program. Most program incompatibilities are usually overcome by this flexible system, though there are some exceptions. Understanding all these options is part of your and my learning curve.

Hardware Resources

Now I want to touch on a reason why people may hesitate about going to OS/2 – the hardware resources required. This topic is divided into hard-disk space, random-access memory (RAM), device drivers, and speed.

Hard-Disk Space: OS/2 requires almost 50 MB of hard-disk space if you install all of it, but only 20 MB for a minimum installation. (30 MB is more typical, says the newsletter's editor.) I suggest that you do a full installation until you know more about the system, and you determine what you don't really need.

In comparison, Windows NT, the only Microsoft operating system able to take advantage of the 80386 and 80486 chips, requires almost 100 MB of hard-disk space!

The key was to put an operating system on a partition separate from your applications.

The Windows alternative uses over 18 MB on my hard disk for just the Windows 3.1 and outdated MS-DOS 5.0 programs. Plus, to match OS/2's standard features, Windows and DOS need more hard-disk space for QEMM, a print cache, and outstanding multimedia capabilities.

If disk space is in short supply, then a simple DOS operating system is your most efficient choice.

If you are currently using DOS or Windows, and you intend to install OS/2, I recommend reformatting the drive or partition onto which OS/2 will be loaded. I have separated my single hard disk into three partitions:

- The C partition holds DOS, Windows, and the programs and drivers needed to make these work better.
- The D partition is for applications.
- The E partition holds only OS/2, and is formatted in the old standard DOS FAT format.

This arrangement allows me to take advantage of the OS/2 Boot Manager to select between DOS/Windows or OS/2 when booting the computer. It reduces any concerns if (actually, when!) I later switch to the faster OS/2 High Performance File System (HPFS) format, or I choose to delete native DOS and Windows entirely.

This kind of partition arrangement was first suggested to me by IBM. The key was to put an operating system on a partition separate from your applications. Then, if there is an update of the operating system, installing the update will never affect all the applications that you have already loaded and set up. (This is a good idea even for DOS/Windows users.) Also, consider yet another partition for your data.

RAM: Though the OS/2 product box suggests that OS/2 will run with a minimum of 4 MB of RAM, most IBMers suggest having at least 6 MB, and preferably 8 MB, to run OS/2 2.1 at a reasonable speed. A key piece of information is that OS/2 2.1 itself wants about 2 MB just to manage and control the computer.

New releases of OS/2 are planned to operate as well in 4 MB of RAM as OS/2 2.1 now operates in 8 MB. Watch for real-world reports.

The less RAM you have, the more often OS/2 uses the relatively slow hard disk to swap files around. But Windows does the same thing, and real-world Windows users talk about needing a minimum of 4 to 6 MB of RAM. Even under DOS, many programs make use of extended memory to speed up operations. And Windows NT suggests a minimum of 12 MB of RAM!

Device Drivers: Device drivers are programs that tell an operating system how to interface with a piece of hardware. For example, when you run a CD-ROM under DOS, a DOS driver for that particular CD-ROM is needed in your computer's CONFIG.SYS file. Printers, tape

backups, scanners, and video cards also need drivers.

Many DOS and Windows drivers work under OS/2 when in an OS/2 DOS or WIN-OS/2 session, but many may not. The OS/2 package comes with a number of drivers. More OS/2 drivers are being written every day, but many specific peripherals are not now supported.

In a manner unlike DOS or Windows, OS/2 takes positive control of all of your computer's hardware. Any non-standard or out-of-spec hardware or device driver that OS/2 finds will be dealt with in varied, usually non-catastrophic, ways.

The moral of the story on device drivers is: If you now use hardware that requires a driver, call the manufacturer and ask if they have an OS/2 driver. Second, talk to IBM either on CompuServe or at 1-800-992-4777, and ask if you can use your specific DOS or Windows driver in OS/2 in a DOS or WIN-OS/2 session. Third, if you plan to buy something, make sure there are OS/2 drivers so that you can take advantage of all the capability of your computer and operating system.

Speed: I haven't had much personal experience in comparing different CPU chips, but everything I've read suggests that there are major improvements to OS/2 operations with the faster chips. OS/2 will run on a 386/486 SX or DX computer, but not on a 286. Windows NT does not run on a 386 SX.

I recommend adding a math co-processor if you have a 386 SX or DX, or a 486 SX. A math co-processor is used during monitor screen refreshing and updating activities, as well as for spreadsheet math functions. Since OS/2 is a Graphical User Interface (GUI) system, anything that speeds up the monitor is good.

The monitor is often the bottleneck to realizing the full potential of OS/2. This is also true for DOS and Windows. That speedy 386/486 chip sits and waits while a program creates a screen "picture," and then "draws" it on the monitor. Video cards are getting faster, and new bus systems (local VESA and PCI) make a significant difference. Just be sure that the video card has an OS/2 driver.

Applications

Another reason some may shy away from OS/2 is the perception of a dearth of applications that take advantage of the unique aspects of the OS/2 operating system.

On the plus side, the several OS/2 magazines show monthly increases in advertising about new OS/2 applications, and several merchants now specialize in OS/2 applications. Based on what I've seen demonstrated, programs written for OS/2 can be very fast, and they offer capabilities not even possible under DOS or Windows.

Even today, OS/2 can run your existing DOS and Windows programs. One estimate is that there are 50,000 DOS applications, 7,000 Windows applications, and 1,200 OS/2 applications (many of these unique to specific businesses). Since OS/2's DOS runs DOS programs better than the standalone DOS you're now using, and Windows programs run at least as fast, you could say that OS/2 2.1 really has over 57,000 applications now on the shelf!

What can be done to encourage development of 32-bit OS/2 applications?

On IBM's part, they operate a BBS and they sponsor several CompuServe forums where you can get questions answered, ask for advice, and download new drivers and hundreds of shareware/freeware programs.

IBM also helps user groups by sending information about new applications. In fact, feedback and interest from local user groups and SIGs can help convince other software companies to produce 32-bit OS/2 applications.

More application and utility programs are needed in order to improve existing products to take advantage of the unique capabilities of OS/2. The prices of OS/2 applications could also use some stiff competition.

I believe the OS/2 application shortage will be overcome, because OS/2 and OS/2 for Windows will prove to be so useful, so flexible, and so stable as to become popular home/desktop choices.

Least Expensive Improvement – OS/2

You have already paid for your 386/486 computer. OS/2 2.1, today, takes advantage of that 386/486 capability you paid for. Since OS/2 can run up to 240 DOS and Windows programs at the same time (depending on hardware resources), no new application software investment is required until you choose to make it.

Most important, OS/2 can run your existing DOS programs a lot better than DOS, and Windows programs run just as fast and as easily (or better) under OS/2.

The least expensive way to improve the performance of your computer is with software. A willingness to learn OS/2 will, in the long run, make your computer efforts more enjoyable for years to come. Install OS/2 today!

***Len Zakas**, who resides in Camarillo, California, is a computer and management instructor. He actively supports the efforts of all the computer user groups in Ventura County. He is vice president of Assisting Children to Excel, a non-profit organization that provides computers and computer training to children from low-income families.*

Launching Objects on the OS/2 Desktop

Bruce E. Högman
Electronic Data Systems Corporation
Boca Raton, Florida

This article discusses different methods of launching applications by using the OS/2 2.1 Workplace Shell (WPS). Then, using these methods, the article offers different approaches to customizing the OS/2 2.1 desktop both during and after installation.

It addresses using applications of all types, both on an individual workstation and across a LAN, showing how OS/2 desktop objects can be defined to easily access applications that reside on either OS/2 LAN or other LAN systems.

Sections in this article address PM session window creation and message processing, as well as WPS API function calls. The section on Window function calls, WinCreateObject, contains information about setup strings for objects. The section on wpSetup contains detailed information about some setup-string keyword value pairs.

Also discussed is my public-domain program, PMSW, which is a support mechanism for launching objects of different session types, as well as a means of synchronizing execution of different sessions at a high level. PMSW uses the entries in the PM desktop's Window List (also known as the "switch list" in the PM programming documentation) to determine which sessions are running.

This article is based on inquiries from several OS/2 customers who wanted to install standardized desktops on all of the computers in their environments. Some of the REXX programs described in this article are actual software solutions to real problems.

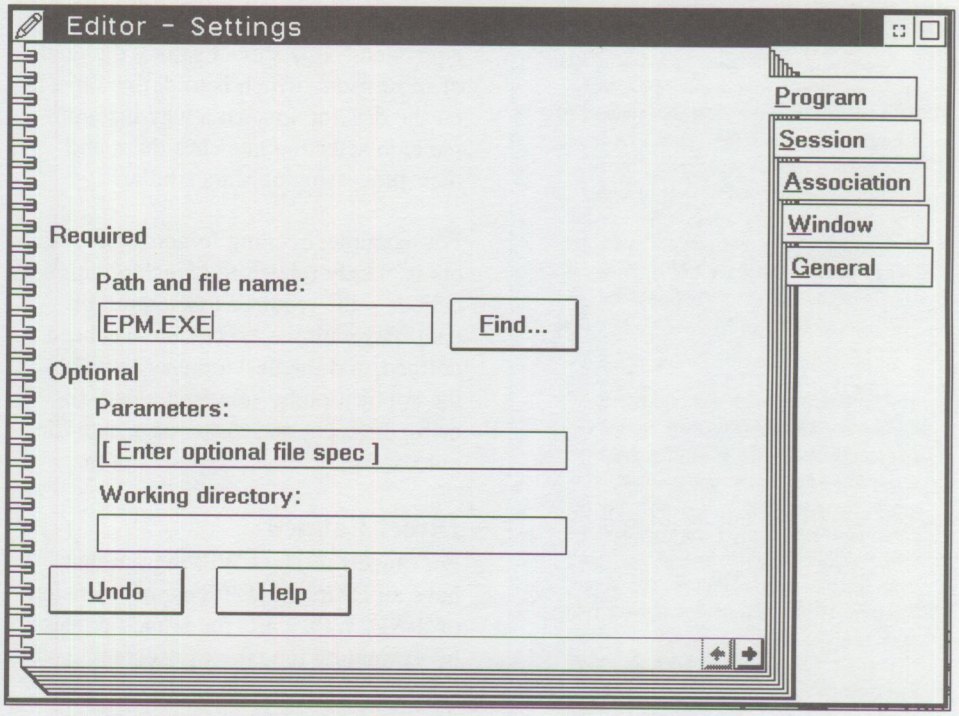


Figure 1. Program Tab Page in Settings Notebook

Launching Mechanisms

The OS/2 desktop provides a Graphical User Interface (GUI) way to launch applications. A user opens an application simply by double-clicking the mouse on an icon associated with an executable. (In this article, this method of launching applications is called *icon selection.)* What lies behind the desktop icon matters little to the user, who cares only that the application starts and runs correctly.

In the Settings notebook for each program object of class WPProgram in the Workplace Shell, there is a Program tab page. On this page, the path and filespec entry-field value can be an executable file (with an .EXE extension) or batch file (.CMD or .BAT extension). The value for path and filespec may be just "*" denoting the default command processor for that type of session.

The Settings notebook presents the object's definition as a number of pages with tabs. Each page contains a group of related parameters. The Settings notebook's Program page contains the path and filename of the program, any program parameters, and the directory to assign as the current directory. Figure 1 illustrates the Settings notebook's Program tab page.

Internally, the path and filename string in a program object's Settings notebook is called an EXENAME setup string.

Regardless of the EXENAME in the Settings notebook, the components of the launching mechanism can be tailored to present the same appearance to the user who does icon selection. Why tailor the launching mechanism? It is easier for everyone if the application designer sets up an icon on the desktop for the user to select, instead of telling the user to "Open a DOS command-prompt window by selecting . . ." and then "Issue the command . . .". If the application designer has done the packaging work well, then the user simply selects the icon, and the possibly complex processing mechanisms launch the application.

References in this article to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service.

The information contained in this article has not been submitted to any formal IBM test and is distributed as-is. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The most common type of launching mechanism is the one in which the path and filename in the Settings notebook point to the executable that is launched. Let's call this a *direct* launch. In contrast, an *indirect* launch uses REXX programs and other methods. Later, this article shows how REXX can extend the functionality of a WPProgram object on the desktop.

There is another way to start programs: invoke them from the command line in a DOS or OS/2 window. This article does not discuss the command-line method, except in situations where a command is issued as part of a launch mechanism that is based on icon selection.

Whether the direct or indirect launch mechanism is used, the user who uses icon selection should see no functional difference in the way that the program is started. As long as the application starts in the same way, all methods of defining and launching the application are equivalent. This should satisfy the goal of most users, which is to define objects on the desktop in such a way that simple icon selection launches them and their processing appears similar.

For example, opening folders or containers is clearly related to launching applications – the required association of data files with the executable has been defined, and the desktop user launches the application by selecting a data file, or by dragging-and-dropping a data file onto an application program's icon.

Direct Launch

For objects of class WPProgram that have an EXENAME filename extension of .EXE, WPS starts the launch process by examining the type of executable – OS/2, DOS, or Windows. In addition, the session can be windowed, full-screen, or PM. Windows sessions, invoked through WINOS2, can also use different Windows run modes, either Standard or Enhanced 3.1.

The launch process for an executable object consists of these general steps:

- Start a process to control the executable
- It sets up the executable environment
- It loads the target executable
- It passes control to the executable
- Which runs within its execution environment
- Which terminates after running
- The control process cleans up the environment
- The control process terminates

These steps do not account for the technical details that differ for each type of executable environment, but rather they try to show the functions performed that are common to all types of executables. The steps outlined can be mapped onto several different operating systems, including IBM MVS for mainframe systems.

The first and last steps above are accomplished by mechanisms in the operating system (OS), which is the overall control process for the computer system. The OS cleans up resources after the application control process terminates.

The command processor that is the control process depends on which type of session is being started. You can start a Windows application from an OS/2 command prompt window or full-screen by issuing a command such as:

```
winos2 appname
```

where **appname** is the program name of the Windows application to be run when WINOS2 starts a Windows session.

Figure 2 shows the default command processors used to control each type of session.

Program Parameters: In the Settings notebook's Program tab page, the Parameters value for a WPProgram object can be set either to accept data input by the user in response to a prompt, or to accept data from a drag-and-drop operation in which a data file is dropped onto the icon of the WPProgram object.

A Parameters value that has matched square brackets surrounding a prompt string will prompt the user for keyboard input. For example, if the string [Enter filespec] is in the Parameters entry field, the desktop displays a dialog box in which the prompt string appears, followed by a data-entry field into which the user types data. (Had the user instead launched a program from a command-line prompt, the user could also have supplied the data immediately after typing the program's name.)

Special Parameters: Figure 3 shows some special string components that go into the Parameters entry field of the WPProgram object. These components

are used to insert varying information (given in Figure 3) about the path, filename, and extension into the command line that invokes the target program. These string components are useful for drag-and-drop operations, because they provide the target program with values that have been prepared and parsed by the operating system.

For example, consider this Parameters string:

```
%*   %**P   %**N   %**E
```

This Parameters string, when used with a drag-and-drop operation, results in a command line that contains:

- the fully qualified filespec (drive, path, filename, extension), from the `%*` string
- the path with last backslash, from the `%**P` string
- the filename with no extension, from the `%**N` string
- the extension, from the `%**E` string

This Parameters string makes the operating system do some of the work, breaking out the parts of a filespec before the application program looks at the command-line operands.

Parsing a filespec can be complex when HPFS is used and the user constructs directory names or filenames that include spaces. A simple technique for handling HPFS names is to construct a Parameters string similar to the one shown above, along with additional special characters used as punctuation. Using such a Parameters string, the resulting command-line operands can be parsed successfully by REXX or by an application program. Here is an example of such a Parameters string:

```
(%*)  (%**P)  (%**N)  (%**E)
```

Let's put this string into action. Consider what happens when a user drops a file named "A file name!.xtx" contained in a directory "A Dir Name" onto a program object that has the above Parameters string. The resulting command line contains:

```
(c:\A Dir Name\A file name!.xtx)
(c:\A Dir Name\) (A file name!)
(xtx)
```

An application program or REXX can parse this command line by using the punctuation added around the special strings.

Indirect Launch

The indirect launch method can be described as using a "helper" mechanism to configure the execution environment before starting the target application. In cases where the target executable requires special handling or environment values that are not in effect for the OS/2 system as a whole, the required environment can be configured using .CMD files or REXX programs. The same is true for DOS applications, where a special .BAT file or AUTOEXEC.BAT that is tailored for the target DOS application can run first to configure its special environment.

The indirect launch method using command files or REXX programs can run applications (programs, REXX programs, or command files) in either a sequential or parallel manner.

If the batch file or REXX program running in an OS/2 session invokes the target program by specifying only its name on a command line, then the target application runs while the invoking batch file is blocked, awaiting termination of the target program. This is called sequential execution of the invoked application.

If, however, the batch file invokes the target program using the START command (see the later section "START Command Syntax and Use"), then the target program will run in parallel with the batch file. This important concept is covered later in the section "PMSW – PM SWitch Desktop Focus."

Session	Default Command Processor
OS/2, PM	\OS2\CMD.EXE
Windows	\OS2\MDOS\WINOS2\WINOS2.COM
DOS	\OS2\MDOS\COMMAND.COM

Figure 2. Sessions and Their Default Command Processors

String Component	Description
`%*`	Fully qualified path of the file
`%**P`	Path with last backslash (except at root)
`%**D`	Path with ':' or Universal Naming Convention (UNC) name (see Note)
`%**N`	Name without extension
`%**F`	Name with extension
`%**E`	Extension without period

Note: A Universal Naming Convention (UNC) name is a name given to a device, server, or resource to give users and applications access to the resource across the network. An example is \\server\drive\file.ext .

Figure 3. Special String Components in Parameters Entry Field of WPProgram Object

In OS/2 2.1, the DOS Settings notebook supports use of an AUTOEXEC.BAT file or another batch file in the DOS_AUTOEXEC session settings. Although the file can have any name, AUTOEXEC.BAT is used here.

The indirect launch category includes the use of an AUTOEXEC.BAT file because a special AUTOEXEC.BAT is used to tailor the runtime environment. The command processor establishes the DOS session environment, and then launches either the batch file specified by DOS_AUTOEXEC or the default AUTOEXEC.BAT before launching the target DOS executable. The AUTOEXEC.BAT can also launch the target DOS application by invoking it as a command or via the CALL command to another .BAT file.

In the case where AUTOEXEC.BAT invokes the target application, the Settings notebook's path and filename entry contains "*", meaning the DOS command processor. The AUTOEXEC.BAT file contains a command line that invokes the target application, and then the EXIT command for closing the command processor's window. Both the AUTOEXEC.BAT and the notebook's path and filename entry can invoke programs or batch files, which means that more than one process can be run, sequentially.

Execution environment requirements for some applications can include:

- Extending or changing the default PATH value
- Changing the DPATH value (in OS/2 applications)
- Running several command, REXX, or program files
- Conditional execution of programs
- Coordinated execution of applications

The most sophisticated and powerful method of indirectly launching an application is to create a program object using System Object Model (SOM) methods, and to launch it during object creation by specifying OPEN=DEFAULT in the WinCreateObject setup string. The current REXXUTIL function package in REXX provides the SysCreateObject function to create objects of all types for the desktop or for desktop folders. See the later section "Setup Strings for Programs" for details about the values in object setup strings.

The current REXXUTIL function package in REXX provides the SysCreateObject function to create objects of all types for the desktop or for desktop folders.

One of the advantages of creating an object for launching applications is that the object can be left visible on the desktop and can appear to be an ordinary application. The mechanisms used internally to launch and control the target application can be hidden from the user, who need not be concerned with those mechanisms. (The internal mechanisms are discussed later in detail.) The object created for launching may also be hidden easily by using the location <WP_NOWHERE>, which is the OBJECTID of the \NOWHERE folder, as shown in Figure 16. Figure 16 contains a REXX program that creates an object for immediate launch, and then hides that object by specifying the location <WP_NOWHERE>.

When creating a program object to launch an application, the launch method can be either direct or indirect. An example of a REXX program that creates an object to launch an application is covered in the next section.

Using REXX programs to create objects provides the ability to query the current operating system environment, attached LAN network resources, and so on. Some of the advantages in using REXX in this manner are:

- Setup can be changed dynamically
- Internal launch mechanisms can be changed
- Objects can be hidden easily
- Objects can start different types of sessions
- Objects can be displayed on the desktop or in folders

Sample SysCreateObject Launch: Figure 16 lists a REXX program for creating an invisible object and launching it upon creation. This program uses PMSW to switch focus to the session if it is already active.

To make this object visible on the desktop, change the Location value in Figure 16 from <WP_NOWHERE> to <WP_DESKTOP>. The location also may be any folder for which you can identify the OBJECTID, or a path such as C:\DESKTOP.

If you make the created object visible, then you don't need to run this REXX program to launch it. Instead, just select the icon (the visible object) to open an OS/2 command-prompt window, maximized. To open a new windowed command prompt each time you select an icon, change to CCVIEW=YES; in the setup.

Sample DOS Object Launch: The REXX program in Figure 17 creates an object on the desktop that runs DOS QBASIC** in a maximized DOS window.

This REXX program creates a visible object that you can launch by icon selection. Using this approach, you can use

the graphic mouse cursor to navigate within this DOS application.

Once you run this REXX program, you don't need to run it again, although you can re-run it to recreate the object on the desktop if you previously dragged it to the shredder.

You can use this REXX program to try new things, such as making the object undraggable.

Launch by Association

An application can be defined to the OS/2 Workplace Shell by using the Settings notebook's Type and Association pages. These techniques require defining data in the Settings notebook for either the application program or for data files to be used with the program. When the association of data files with a program is done correctly, the user can invoke the target program to operate on a particular data file. To do this, the user selects the icon of the data file that is displayed in an open container view.

There are three ways to set up a data-file object so that a program is started when the user selects the data file's icon. These methods require updates to the Settings notebook for either the application program or for individual data files. The updates can be done manually or by a program. These methods are:

- Create a filename association in the application program's Settings notebook (as shown in Figure 4).
- Create a type association in the data file's Settings notebook (Figure 5).
- Make the program name the default setting in the "Open" cascade menu in the data file's Settings notebook (Figure 6).

An association is a special link that can be assigned to a program object. This link allows the user to specify which program will be invoked when the user double-clicks on specified types of files,

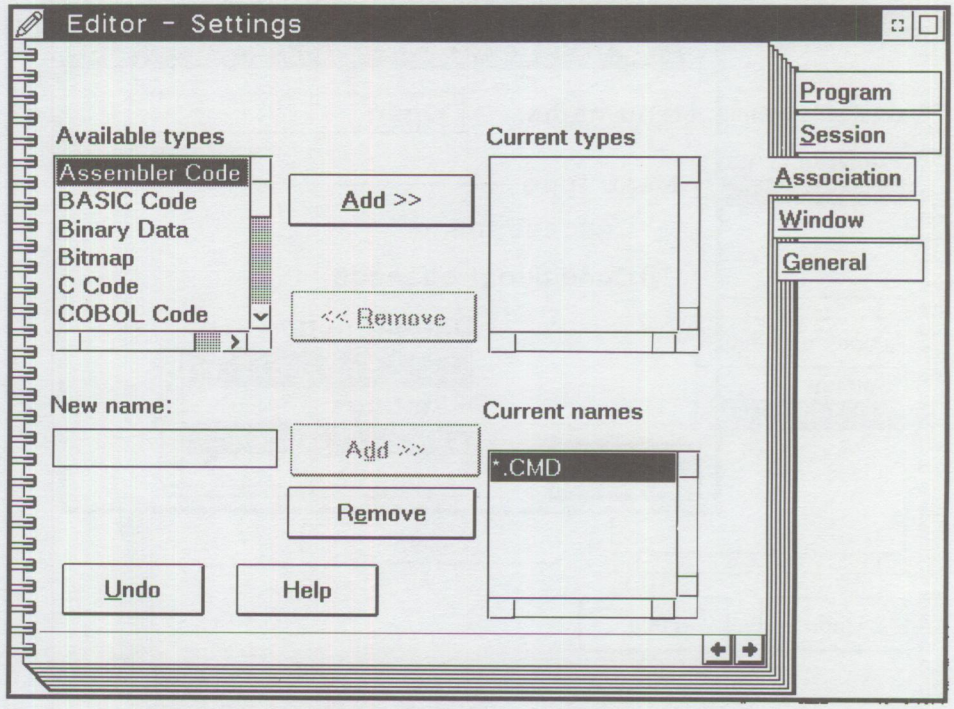


Figure 4. Program-Object Association Page

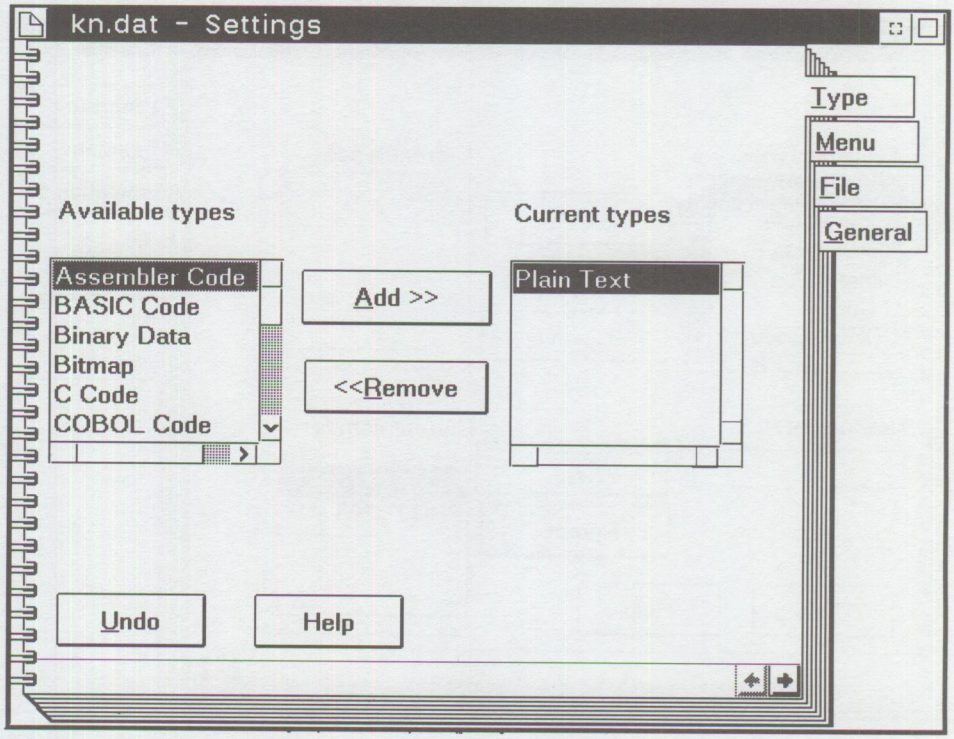


Figure 5. Data-File Type Page

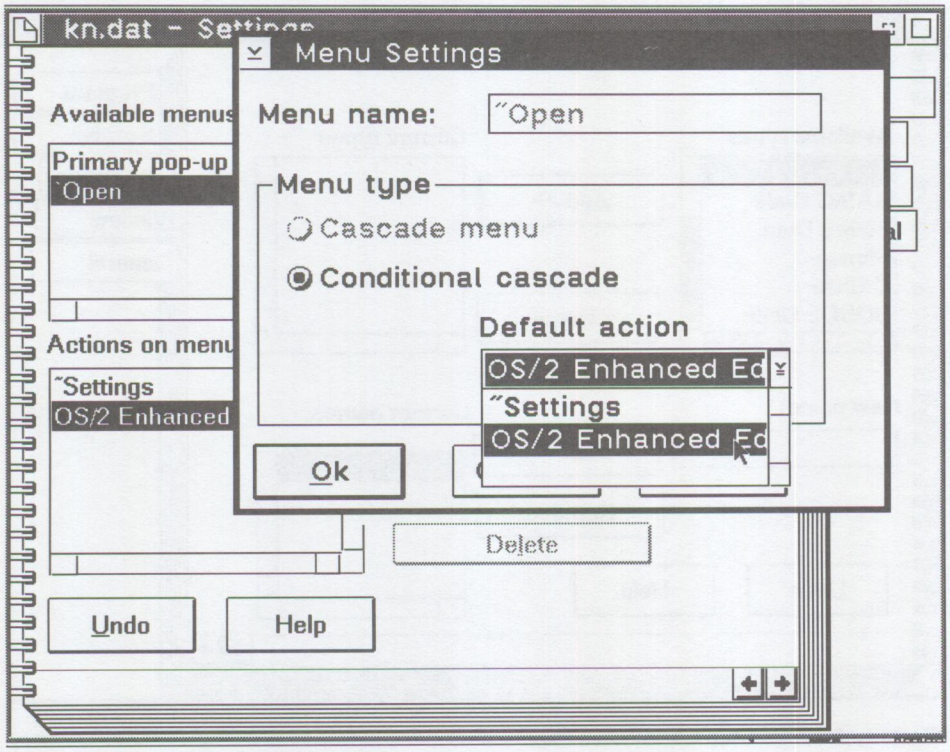


Figure 6. Defining New Default Action

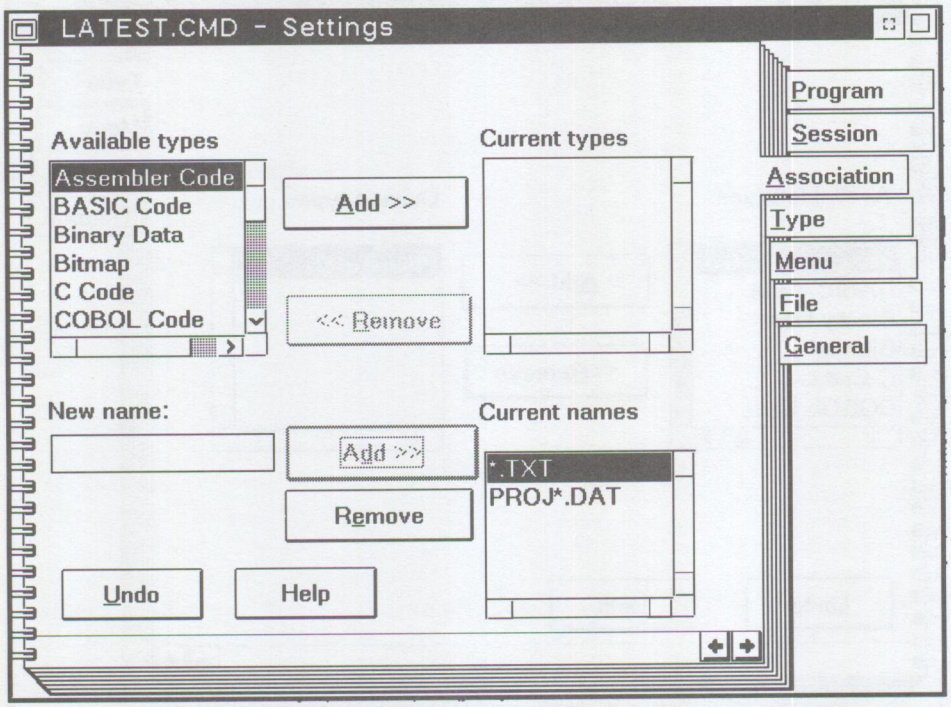


Figure 7. File Name and Extension Association

individual files, or files described by a global filename mask.

Filename Association in a Program: Creating a filename association in a program's Settings notebook requires that parts of filenames remain constant, so that data files can be associated with the program. Most associations reserve a file's extension, although the part of a filename that must remain constant does not have to be its extension. This reservation of part of a filename is not very flexible, and associations can be destroyed accidentally by changing the data file's filename or extension. This method supports the use of HPFS long names.

To create an association between a program and data files, open the Association page in the program object's Settings notebook. Enter the filename mask or filename extension in the New Name entry field, and select the Add pushbutton to define the association. The "?" and "*" characters can be used in the filename mask or in the extension.

Figure 7 gives an example of setting an association based on part of a filename and also a file extension.

A text editor may, for example, have associations for plain-text files (that is, a type of data file), the individual file CONFIG.SYS, and all files described with the global name READ*.*. Whenever a data-file object that meets these criteria is selected with a double-click, OS/2 starts the appropriate program object.

The filename-association method is not very flexible. When a user using icon selection wants to use a different program with the data file, the existing association must be changed. This is usually too complicated for the average user, and it involves too much manual work to be efficient.

Type Association in a Data File: Creating a type association in a data file's Settings notebook is a better approach. The advantage of this approach is that it is easy for a user to use a different program with the program object, simply by choosing a new type in the Settings view. The process for making a type association in a data file is simple:

- Open the Settings view for the data-file object
- Select the Type page
- Select from the list of available types
- Press the Add button
- Close the Settings window

Individual programs can be associated with a Type value, but support is limited, and usually only OS/2 programs support file-type association. Programs can also create new file types, and they can create the data-file associations during processing, to provide the GUI application launching for users.

Type association in data files is described in more detail in *Presentation Manager and Workplace Shell Application Development* and *Workplace Shell Implementation* in the references below.

Adding a Default Program to a Data File's Menu: Adding a default program to a data file's menu should be undertaken only by users who understand the workings of the Workplace Shell in some detail. Its main disadvantage is evident when many existing files must be set up to use a particular program: each one must be modified separately.

If you are setting up a system for a new user, you can create a data file with a particular program identified as the default in "Open" in the context menu, then make that file a template so that all new files subsequently created from it will inherit this default attribute.

To create new files from the template, drag the template with the mouse. Dropping the template creates the new instance of the data file, with all of its associations and context menu items intact.

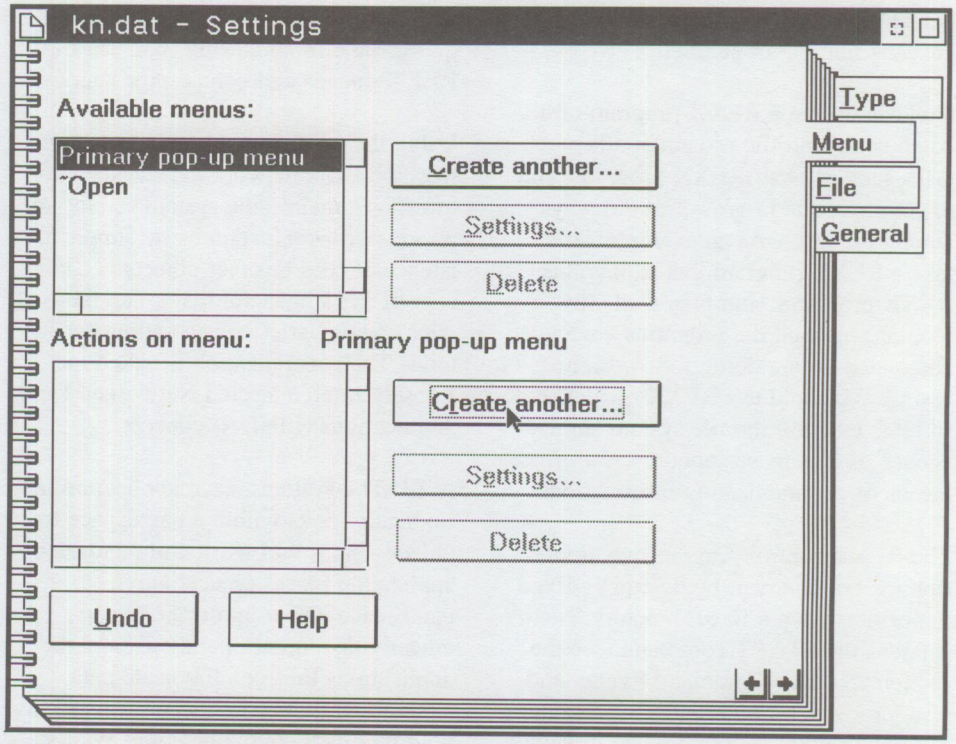


Figure 8. Adding a New Menu Item

The *PM and Workplace Shell* Redbook contains detailed instructions for updating an object's context menu. This is the menu that you display when you click mouse button 2 (usually the button at the right) on the object's icon. You can add new menu actions, and you can define a new default action that is taken when you select the Open option in the menu.

Figure 8 shows how to add a new menu item to the notebook menu page, and Figure 6 displays the pop-up menu that you see when you select Open, then Settings, to define the new default menu action.

Coordinating Applications

The focus of this article thus far has been on methods of launching applications. It now switches focus to cover methods of coordinating processing among applications.

Gross Methods of Task Coordination

A batch file running in a DOS session can invoke programs or other batch files sequentially only. A batch file or REXX program running in an OS/2 session can invoke other sessions either sequentially or in parallel, depending on the command used to launch the invoked sessions.

If an OS/2 CALL command is used to invoke a program, the invoked program will execute while the OS/2 session that invoked it is blocked, waiting for completion of the program. The abilities of the CALL method are limited, however, and the invoked program runs with the global default processing environment.

It is often useful, and sometimes required, to use more powerful methods of invoking programs. These methods usually require use of the START command or other launch mechanisms. In turn, this introduces new problems of

managing events among the tasks that are now running in parallel.

You can design a REXX program or a batch command file to launch other tasks such as other REXX programs and to communicate in gross (rough) ways among the tasks. As an example, let's take a REXX program that starts other REXX programs, and then waits (by looping) until all the programs have completed before doing follow-on processing. We could use REXX queues, but instead, let's use the file system and existence of files as semaphores as a gross means of communicating information.

The REXX control program sets up the runtime environment by deleting a fixed list of files from a fixed directory. Next, it issues the START command (see the section "START Command Syntax and Use") for several other REXX programs or DOS or OS/2 sessions. Then it waits for a while by using the SysSleep function in the REXXUTIL package.

After it awakens, the REXX control program can use SysFileTree to build a list of filenames to see whether the started sessions have completed yet. As each of the started sessions begins, it creates a file, whose name is fixed, which indicates that the session is active now. As each session ends, it deletes the file it first created, then creates another file that indicates it has completed. The REXX control program looks for these files, and can thus tell when each session terminates normally. However, if one of the sessions terminates abnormally, that situation may not be apparent to the REXX control program.

The above example is gross in its design, its interprocess communication, and its exploitation of system services. Proper use of REXX features can make this simple example much finer. If you want to create a task environment in which parallel processes communicate quicker and more cleanly, you can use REXX queues and named-pipe support.

Using the operating system's file services, the LAN network namespace features, or a mainframe system catalog to provide gross semaphores by simple existence of files or other objects is a concept that is so general that it can be used over a wide variety of operating systems. The generalization is called the Loosely Interconnected Networked Data Applications (LINDA) concept.

In LINDA systems, each application periodically looks within a namespace for objects that it will work with, and each application places objects into a namespace. Other applications periodically do similar operations, and the applications thus communicate data among themselves in a loose network with no direct communication. We use this namespace concept daily when we look up someone in the phone book – a common namespace for people in a locality – or call directory assistance.

REXX programs to search for directories and files across drives using parallel REXX program tasks are available in the author's RXMP package on CompuServe OS/2 forums and the IBM BBS. These programs use the RXQUEUE function to coordinate task processing and to pass data from child task to parent program.

PMSW – PM SWitch Desktop Focus

The PMSW program, used in REXX programs to communicate the existence of active tasks by name, can be thought of as a type of LINDA application. The task that was launched by icon selection or by START is listed in the desktop's Window List.[1] PMSW is told to look in that list for a particular name, and it then either switches desktop focus to that task or it returns a code indicating whether the task is or is not active.

This PMSW.EXE program or its REXX external function (PMSW2.DLL) can be used in batch files or REXX programs that launch a target application in parallel execution mode. To do this, the REXX program that uses PMSW.EXE or PMSW.DLL should either use the START command or launch a program object by creating or updating it with the OPEN=DEFAULT; setup string. In this article, only REXX programs are discussed, because only REXX has the needed functionality to exploit the PMSW program.

When a batch file or REXX program issues a START command, the START command starts another session that runs in parallel with the batch file or REXX program that issued the START. The launching REXX program continues to run, and can do other processing after having launched the parallel process. The launched session appears in the desktop's Window List when it becomes active and can receive the focus.

PMSW gives a REXX program the ability to launch a session and then switch active focus to that launched session, regardless of the current focus, and regardless of whether the launching REXX program is in the foreground or background.

PMSW can be invoked from batch files or REXX programs running in foreground, background, and minimized sessions, or in sessions launched by the START command. Even though PMSW is a PM application, it can be used within a process started by the OS/2 command DETACH. A process started by DETACH cannot acquire the focus due to its lack of PM resources. By definition, DETACH starts a process that

[1] The term Window List is used in the *OS/2 Command Reference*, and that is indeed the title of that window. However, programming references and internal code refers to it instead as the switch list. This article uses Window List.

does not use keyboard or screen resources. PMSW does not use keyboard or screen resources to switch focus, so it can be used with DETACH.

PMSW provides the mechanism by which a "parent" REXX program can track the independent "child" process through the presence or absence of the child's task name in the desktop's Window List. To be able to do this, the REXX program must know, in advance, the name of the child process. A name can be given to the child task by using the START command. (The syntax of the START command is illustrated in Figure 11 later in this chapter.) The started session can be named by coding the optional first operand within double-quotes. PM programs started in this way may change their Window-List names by making an appropriate API function call.

Using PMSW, a REXX program can check to see if a given task is active in the Window List and, if so, the program will wait by using the REXX SysSleep function, then check again after SysSleep. The watching REXX program can then delay further processing until the watched task terminates, or do other processing in parallel while the watched task is still active. A REXX program that uses PMSW to watch active tasks can do a list of tasks sequentially by starting a task, watching it until it completes, and then start the next task in a list. Similarly, a REXX program that uses PMSW can launch a number of tasks in parallel, watch for the last one to complete, and then do something else, such as cleaning up anything left by any of the tasks.

PMSW Version 2 now supports "switch to self", so a REXX program that is running and that uses PMSW to switch focus to another session can later return focus to its own session.

Detailed information with sample REXX programs is given in the section "PMSW Function and Command Line Syntax."

```
RUN=C:\OS2\XCOPY.EXE C:\OS2\OS2.INI      C:\OS2\INSTALL
RUN=C:\OS2\XCOPY.EXE C:\OS2\OS2SYS.INI   C:\OS2\INSTALL
RUN=C:\OS2\XCOPY.EXE C:\CONFIG.SYS       C:\OS2\INSTALL
```

Figure 9. RUN Commands for Backing Up Critical OS/2 Files

Desktop Customization

The standard OS/2 2.1 installation processing creates an initial desktop during the last step of the installation, when the system is booted. A starter set of desktop folders and icons is constructed by the Workplace Shell during the first boot process. The user sees these standard objects, as well as the OS/2 Tutorial, which is presented automatically.

Users want icons on their desktops for the applications that they use the most. They can manipulate the desktop using the mouse to drag-and-drop, create shadows, etc. If a company wants to distribute OS/2 2.1 and to create objects on the desktop or in folders for all users, it is usually convenient to do so during installation of OS/2 2.1. The objects that a company creates during customization can either augment or replace the standard objects.

Indeed, this is a topic of interest across the industry – how to mass-produce a desktop that has a set of applications already defined for users without going to a lot of trouble. There are several ways to do this:

- Use REXX SysCreateObject
- Change the contents of the file \OS2\INSTALL\DATABASE.DAT
- Use MAKEINI to update the base OS2.INI file, the user file that contains definitions of objects.

Each of these ways is discussed in sections that follow, with some recommendations as to when each method would be best.

Saving the Desktop Customization: Another hot topic for all users is how to save the desktop in anticipation of the day when the OS2.INI file gets destroyed.

Users can press keys Alt+F1 at boot time to cause OS/2 to copy the files from \OS2\INSTALL to the \OS2 directory and to the root directory. These files are CONFIG.SYS, OS2.INI, and OS2SYS.INI. If you don't want to use the Alt+F1 restore process, you can also boot from the installation disks, or from bootable OS/2 diskettes, to manually create backups of the critical files.

One method of creating a backup copy of the critical files that can be restored easily is to include RUN commands in the CONFIG.SYS file. These RUN commands copy the OS2.INI and OS2SYS.INI files, and CONFIG.SYS itself, to the \OS2\INSTALL directory.

Figure 9 illustrates this concept. In Figure 9, the boot drive is assumed to be C.

This simple copying may cause problems. If an error exists in the CONFIG.SYS file that causes failure of the Workplace Shell, the RUN command processing has already copied the bad CONFIG.SYS file into the backup directory, so you won't be able to recover using Alt+F1.

A better way would be to copy the files during boot to another directory, say \OS2\BACKUP, and then periodically to manually copy the critical files from \OS2\BACKUP to \OS2\INSTALL. This staging of the backup files avoids overlaying the files in the \OS2\INSTALL directory each time you boot.

If you have sufficient hard-disk space, you can copy the critical files to the \OS2\BACKUP directory and then run a batch file from STARTUP.CMD to do the following process:

1. Erase file xxx.003
2. Rename file xxx.002 to xxx.003
3. Rename file xxx.001 to xxx.002
4. Rename file CONFIG.SYS to CONFIG.001
5. Rename file OS2.INI to OS2.001
6. Rename file OS2SYS.INI to OS2SYS.001

This process can be modified further to automate the copy of files to \OS2\INSTALL by copying the xxx.003 version (or whatever is highest, oldest-numbered version) to the \OS2\INSTALL directory with renaming:

7. COPY \OS2\BACKUP\CONFIG.003 \OS2\INSTALL\CONFIG.SYS
8. COPY \OS2\BACKUP\OS2.003 \OS2\INSTALL\OS2.INI
9. COPY \OS2\BACKUP\OS2SYS.003 \OS2\INSTALL\OS2SYS.INI

This modification "buffers" the overlaying of the files that would be restored using Alt+F1 processing during boot.

Using REXX SysCreateObject: Using REXX with SysCreateObject calls to define folders and program objects to produce desktop customization is a generally useful, easy process. This method consists of using REXX programs that call the SysCreateObject function to update the OS2.INI file.

This method has only one prerequisite, the build of the initial desktop. It cannot be used prior to the build of the initial desktop; it is a post-installation method only.

A REXX program that installs objects on the desktop can also create the required application directory and file structure. The REXX program can include code that lets the user choose the drive and path for installation of the application; or, the target can be selected automatically based on available disk space; or, the target can be constant. An alternative is to use a separate installation process, such as CID or NVDM/2, to install the directory structure and files, and then to define the desktop objects using REXX. See the section "SysCreateObject REXX Function" for the syntax and examples of this function.

A REXX program that installs objects on the desktop can also create the required application directory and file structure.

In the call to the function, you include the type of object, displayed title, location name, object setup string, and update or replace option. For desktop customization, most of the objects are folders of class WPFolder, or application programs of class WPProgram.

Among the object setup-string values, you should include a specific, unique OBJECTID value so that you can refer to the object in other REXX programs. By giving a folder an OBJECTID name, you can open that folder automatically in a REXX program.

OBJECTIDs can be used successfully to create and update objects only if the objects are not duplicated by using Workplace Shell functions. Consider the situation where you have created a folder with an OBJECTID, and program objects within that folder, with each program object having its own unique OBJECTID. Then, what happens when you COPY the folder? The program objects within the duplicate folder no longer have OBJECTIDs to which REXX programs can refer.

Designers of desktop customization must be aware that the internal object "handle," assigned by the Workplace Shell when the object is created, is the only fixed, unique object identifier. Since a REXX program can refer to objects only by OBJECTID or by a fully qualified path and filename, the changing of an OBJECTID by WPS processing represents a limitation on the ability of REXX programs to handle objects.

Changing \OS2\INSTALL\DATABASE.DAT: This method uses the MIGRATE.EXE application migration program in OS/2 to tailor the desktop during initial installation or at any later time when the user wants to manually run MIGRATE. This method updates the OS2.INI file after it asks the user to select the programs to include in the migration process.

The disadvantages of this method include the following:

- The need to edit \OS2\INSTALL\DATABASE.TXT
- Possible need to edit \OS2\INSTALL\DBTAGS.DAT
- Much hands-on dialog work with MIGRATE.EXE

The prerequisite for this method is the existence of the directory and file structure required by the application. You must have already created the directories and files by using a command file or REXX program, or else the directories and files must be addressable on LAN resources.

This method starts with examining the DATABASE.TXT file to see if it already contains the target application. If not, then you must copy an existing entry as a template for your new application.

Unless a new application defines new session parameters in its Settings notebook, there is no need to edit the DBTAGS.DAT file. In fact, the file contains a warning not to edit it. In most cases, editing this file has little effect unless the associated CONFIG.SYS driver files have already been loaded; these driver files contain the required resource names for the settings parameters that will be added.

After you have edited the DATABASE.TXT file, you must update the DATABASE.DAT file (which is used by MIGRATE.EXE) by running PARSEDB.EXE in the \OS2\INSTALL directory. After that, you must run MIGRATE.EXE to migrate any applications you want to install as Windows, DOS, or OS/2 applications.

Running MIGRATE.EXE is an interactive process. The program identifies applications as candidates for migration by finding matches for the program names in the DATABASE.DAT file. It then presents you with a list of these applications, and you can select which ones to migrate from the list displayed. The visible result of the migration process is one or two folders on the desktop containing program objects for OS/2 and Windows applications.

Because of the involved processing required, and because of the manual operations required, this approach to customization is not the easiest. From the viewpoint of the system administrator, it also presents problems in controlling the overall process.

Using MAKEINI to Update the Base OS2.INI File: This method uses MAKEINI.EXE to update the OS2.INI file. In addition to creating new objects on the desktop, you can change system settings, such as date and time formats, to conform with local installation standards.

This method of desktop customization can be done at both initial installation time and at any time thereafter.

Prior to the first boot of the new OS/2 system, you can change the initial desktop by editing the INI.RC file, which is used to create the initial desktop. (For some examples of editing the .RC file to change objects or to add objects during installation, see the section "Setup Strings in the .RC File.") After the first desktop is constructed, you can change the desktop by running MAKEINI.EXE with a USERNAME.RC file. This merges the object definitions in the USERNAME.RC file with the existing OS2.INI file.

There are sample .RC files in the \OS2 directory. Their format and content are described later in the next section, "OS/2 .RC File Format."

Distributing customized desktops during initial installation of OS/2 2.1 is easier than adding customization afterward using MAKEINI.EXE. Doing CID, LCU, or NVDM/2 remote installation of OS/2 2.1, you can use a user exit, which is invoked prior to the first boot, to insert new desktop objects into the INI.RC file or to delete unwanted lines.

When installing OS/2 2.1 using CID, LCU, or NVDM/2, you can simplify the updates to INI.RC by extracting INI.RC from the CID directory structure. Go to the \CID\IMG\OS2V21\DISK_9\REQUIRED bundle file, use the UNPACK program, edit INI.RC to add your desired customization, and then repack the updated INI.RC into the original bundle. Details of this procedure are addressed in the NVDM/2 product documentation.

The PACK.EXE program distributed with the Developer's Toolkit for OS/2 can be used to repack bundle files only if all files are extracted and then repacked as a new bundle file. You need the PACK2.EXE program, available

through IBM support, to repack individual files into a bundle for OS/2 2.11 and later releases.

Using MAKEINI and a USERNAME.RC file after the first boot is more difficult. The steps outlined below are one method:

1. Boot using OS/2 installation diskettes or bootable OS/2 diskettes.
2. Copy the OS2.INI file from the \OS2 directory to another directory for backup and recovery use.
3. Run MAKEINI, specifying \OS2\OS2.INI as the first operand and your USERNAME.RC file as the second.
4. Reboot OS/2 from the hard disk.

MAKEINI merges the results of the .RC file processing with the OS2.INI file. The next boot of the system then completes the creation of objects as specified in the USERNAME.RC file. Recovery from errors requires rebooting using diskettes and copying the backup copy of OS2.INI back to the \OS2 directory.

Refer to the *Presentation Manager Reference* for information about API calls that affect the system and user profile files (usually, OS2SYS.INI and OS2.INI). MAKEINI is documented in the *OS/2 2.1 Command Reference.*

OS/2 .RC File Format: Customizing the desktop using MAKEINI involves creating or changing a resource file such as the INI.RC file that creates the stock desktop. To make changes correctly to an .RC file, you need to understand the overall format of the .RC file, and how its individual entries are constructed.

You can invoke the MAKEINI program manually (by running it at a command prompt), or you can code a program. Using a program with the appropriate API calls, you can:

1. Create a copy of the currently active OS2.INI file.
2. Invoke MAKEINI to merge the new file-object data in USERNAME.RC with the copy.
3. Call the PrfReset API to change the name of the active user profile file from OS2.INI to another name.

This technique requires PM programming expertise.

You can do similar processing by using a command prompt and issuing the commands yourself. The steps listed below assume you have included the commands to copy the .INI files at boot time to the \OS2\BACKUP directory.

1. COPY \OS2\INSTALL\OS2.INI \OS2\BACKUP\OS2.BAK
2. COPY \OS2\BACKUP\OS2.INI \OS2\BACKUP\UPDATE.INI
3. MAKEINI \OS2\BACKUP \UPDATE.INI \OS2\BACKUP\USERNAME.RC
4. COPY \OS2\BACKUP\UPDATE.INI \OS2\INSTALL\OS2.INI
5. Reboot using Alt+F1 to activate the updated OS2.INI

This example assumes your USERNAME.RC file and all other files related to backup are in the \OS2\BACKUP directory.

If you want to restore the last active OS2.INI file, reboot using the installation diskettes, then copy the backup copy from \OS2\BACKUP\OS2.BAK to \OS2\OS2.INI and reboot normally.

When MAKEINI is run, it creates application names, keywords, and values in the target output .INI file. The entries created by MAKEINI are known as STRINGTABLE entries. When WPS initializes at the next system boot, it finds these STRINGTABLE entries of names and values, and completes the process of creating new objects by deleting the STRINGTABLE entries that were placed in the .INI file by MAKEINI.

In editing the INI.RC file used to create the initial desktop, avoid changing the required parts of the file. All .RC files must begin with the CODEPAGE and STRINGTABLE statements. The list of blank values should remain as-is.

To change the default values for system parameters such as PM_National iDate format, change the numeric value to correspond to the desired date format.

The STRINGTABLE entries have the general format of:

1. Application name, such as PM_ControlPanel
2. Keyname, such as BorderWidth
3. Value, such as 4

This particular entry would appear as:

```
"PM_ControlPanel" "BorderWidth"
"4"
```

Adding new objects involves adding new STRINGTABLE lines with PM_InstallObject as the application name, and a keyname string that consists of:

1. Title of object
2. WPS class of object
3. Location of object

The title of the object is what appears on the desktop under the icon for the object. The class of object is one of the WPS classes such as WPFolder or WPProgram. Objects are created in the order in which the entries appear in the .RC file, so folders must be created before creating objects that go in those folders. The last part of each STRINGTABLE entry is the object setup string containing keywords that describe the object, such as OBJECTID, EXENAME, PROGTYPE, and so on.

Object setup strings contained in STRINGTABLE entries that refer to paths for files or programs, and that contain a question mark in place of the boot-drive letter, will have the question mark replaced by WPS when it creates a

```
"PM_InstallObject" "Novell;WPFolder;<WP_DESKTOP>;UPDATE"
   "ICONPOS=25 87;OBJECTID=<Novell_Folder>"

"PM_InstallObject" "Netware Tools;WPProgram;<Novell_Folder>;UPDATE"
     "STARTUPDIR=C:\NETWARE;EXENAME=C:\NETWARE\NWTOOLS.EXE;PROGTYPE=PM"

"PM_InstallObject" "Network Printer;WPProgram;<Novell_Folder>;UPDATE"
    "STARTUPDIR=C:\NETWARE;EXENAME=C:\NETWARE\NPRINTER.EXE;PROGTYPE=PM"

"PM_InstallObject" "Install;WPProgram;<Novell_Folder>;UPDATE"
    "STARTUPDIR=C:\NETWARE;EXENAME=C:\NETWARE\INSTALL.EXE;PROGTYPE=PM"
```

Figure 10. .RC File Entries for Novell NetWare

new object. This means that a REXX program that creates STRINGTABLE entries for export to other systems can specify the boot drive for the target system as "?". This is one method of creating an absolute path, but only for objects on the boot drive.

Here is an example of part of an object setup string (as one line with no spaces):

```
EXENAME=?:\MYAPPL\MYAPPL.EXE;
   STARTUPDIR=?:\MYAPPL;
```

See the *Presentation Manager Reference*, in the topic "WorkPlace Object Class Hierarchy," for details about the keyword value pairs appropriate for WPFolder and WPProgram object classes.

Figure 10 lists four entries that you can add to your .RC file to create a Novell NetWare folder on the desktop and to define OS/2 NetWare Requester programs in that folder. The folder's icon will be placed on the desktop 25 percent across and 87 percent up from the lower left corner of the desktop. Note that each entry in Figure 10 should be a single line in the .RC file.

REXX Program for MAKEINI Function: The MAKEINI program is not the only way to alter the user or system .INI file for the purpose of creating objects. You can also update the current .INI file using a REXX program similar to the one in Figure 18. Note that this REXX program adds the new STRINGTABLE entry to the current OS2.INI file. This is a much easier way to accomplish the MAKEINI function than running the MAKEINI program itself.

Using such a REXX program to create an object illustrates what MAKEINI does when it processes the .RC file STRINGTABLE entries. The next boot of WPS will process the new string entries in the .INI file to create the new objects. Figure 19 lists an example of locating the boot drive for OS/2 systems.

Setup Strings in the .RC File: Whether MAKEINI or REXX is used to update the .INI file with STRINGTABLE entries, or whether REXX SysCreateObject function calls are made to create objects directly, all of these methods require knowledge and proper use of the object setup strings processed by WPS.

To make the Tutorial not come up after installation or MAKEINI, remove the line shown below from INI.RC. This line is an example of a STRINGTABLE entry; it consists of three strings, delimited by double-quotes, that appear in one line in the .RC file.

```
"PM_Workplace:InstallAutorun"
"OS/2 Tutorial"
"EXENAME=TUTORIAL.EXE,STARTUPDIR=
   \OS2\HELP"
```

To make the Drive A object not come up after installation or MAKEINI, remove the following line from INI.RC:

```
"PM_Workplace:InstallDiskOnDesk-
   top" "A" "ICONPOS=80 8"
```

To change the desktop directory to something other than Desktop (for example, to 'My Desktop'), replace the Desktop entry in the following line with 'My Desktop':

(Before)

```
"PM_InstallObject"
"Desktop;WPDesktop;?:\"
"OBJECTID=<WP_DESKTOP>"
```

(After)

```
"PM_InstallObject"
"My Desktop;WPDesktop;?:\"
"OBJECTID=<MY_DESKTOP>"
```

This line in INI.RC creates a directory called 'My Desktop' on the boot drive. Note that the space in the title of the folder will be replaced with an underscore if the file system for the desktop drive is FAT (File Allocation Table) instead of HPFS (High Performance File System). The process that creates Workplace Shell objects handles special characters and long names for FAT systems by truncating long names and substituting special characters.

Consider the case of creating one folder with the title "My Applications for Work" and another folder whose title is "My Applications for Home." Each folder is represented by a directory name that follows the naming convention of the FAT file system. WPS cre-

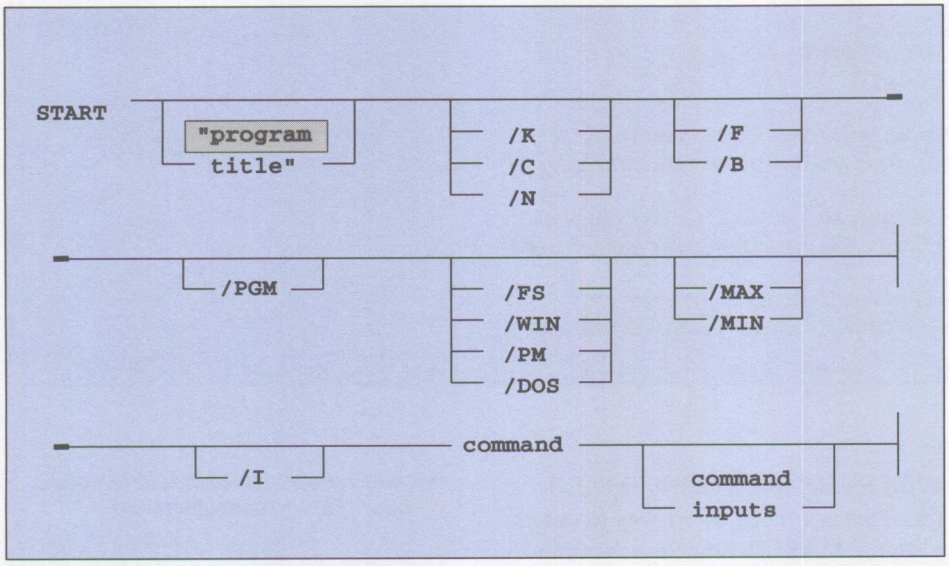


Figure 11. START Command Syntax

ates a physical directory named \MY_APPLI for the former folder, and one named \MY_APPL1 for the latter. The title for each folder is the full name. On the other hand, with an HPFS file system in use, the physical names for the directories are the same as the folder titles.

Note that this method is not good for installations of standardized desktops, and it may confuse users who expect to see \DESKTOP as the desktop folder. However, this method provides a good example and explanation of how the desktop is defined internally, and technical support staff should be aware of it in case they encounter it in dealing with adventurous users.

To create a folder called 'My Folder' on the desktop, add the following one-line entry to the .RC file:

```
"PM_InstallObject"
"My Folder;WPFolder;<MY_DESKTOP>"
"BACKGROUND=?\OS2\BITMAP
   \LIGHTHOU.VGA;OPEN=DEFAULT;
   ICONRESOURCE=61 PMWP;
   ICONPOS=8 70;
   ICONVIEWPOS=18 60 75 22;
   OBJECTID=<MY_FOLDER>"
```

This line creates a folder called 'My Folder' on the desktop. The class of this folder is WPFolder. The location of the folder is the desktop, which has an OBJECTID=<WP_DESKTOP>. The background of the folder will have a lighthouse bitmap. The icon will look like the OS/2 System icon.

When you include OBJECTID in the object setup string in the .RC file, that parameter must be the last one in the setup string.

To add an application to 'My Folder', insert the following information all on one line:

```
"PM_InstallObject"
"OS/2 Window;
   WPProgram;<MY_FOLDER>"
"EXENAME=?\OS2\CMD.EXE;
   PROGTYPE=WINDOWABLEVIO;
   HELPPANEL=8010;
   OBJECTID=<MY_OS2WIN>"
```

To add a shadow of Solitaire to 'My Folder':

```
"PM_InstallObject" "Solitaire
   Shadow;WPShadow;<MY_FOLDER>;"
"SHADOWID=<WP_KLDK>;
   OBJECTID=<MY_KLDK>"
```

To add a shadow of an OS/2 window to the desktop:

```
"PM_InstallObject" "OS/2 Window
   Shadow;WPShadow;<MY_DESKTOP>;"
"SHADOWID=<WP_OS2WIN>;
   OBJECTID=<MY_WINSHADOW>"
```

Documentation about the setup strings is contained in the *Presentation Manager Reference*.

START Command Syntax and Use

The START command starts an OS/2 program in another session. Its primary use is to start programs automatically at system startup. The special batch file STARTUP.CMD allows you to do this.

The syntax of the START command is shown in Figure 11. To imbed redirectional signals into the command session, enclose the command and command inputs in quotation marks.

If you use the /WIN, /FS, or /PM parameter, your program runs in the foreground session. Without using one of these parameters, you can also run the program in the foreground by using the /F parameter.

Make sure that you specify the correct drive and path when you use the START command to run a batch file with the STARTUP.CMD file. Also, if you plan to redirect I/O using the START command, enclose the command and command inputs within quotation marks.

You can use START to run full-screen applications or applications that run in a window, such as Presentation Manager programs.

START determines the type of application, and will run it in the appropriate window or full-screen session. However, you have the option to override the determined default by using the /FS, /WIN, /PM, or /I parameters.

You cannot start a batch file (.CMD) with the /PM parameter.

SysCreateObject REXX Function

To introduce this topic, the text about SysCreateObject from the *REXX Reference* is reprinted here.

The syntax of the SysCreateObject function is:

```
result=SysCreateObject
   (classname,title,location,
   setup,option)
```

where

`classname` is the name of the object class.

`title` is the object's title.

`location` is the object's location, which can be specified as either an OBJECTID (for example, <WP_DESKTOP>) or a file system path (for example, C:\BIN\MYTOOLS).

`setup` is a WinCreateObject setup string.

`option` is the action taken if the object already exists. Allowed options are: "fail," "replace" (delete existing object and create new object), and "update" (update the settings of the existing object). Only the first letter of the option is needed; all others are ignored.

`result` is the return code from WinCreateObject. The return code is 1 (true) if the object was created, and 0 (false) if the object was not created.

When creating objects, you must create the containing object before attempting to create objects within the container. Make the folder first, then create the objects to store in the folder.

Figure 20 lists an example of a program that creates a folder on the desktop and then creates a program object in that folder.

PMSW Function and Command-line Syntax

The PMSW public-domain REXX function interface is available on CompuServe in the IBM OS/2 forums as PMSW.ZIP. The information in this section was extracted from the full PMSW.INF file contained in the PMSW.ZIP distributed version. That file also contains details about technical design, as well as the full C-language source for the PMSW application.

The PMSW application comes in two executable forms when uncompressed:

- PMSW.EXE, an OS/2 PM program
- PMSW2.DLL, a REXX external function

PMSW is invoked with two arguments:

```
PMSW name_mask_string [/r]
```

where

`name_mask_string` is the name mask for the task name in the desktop Window List

`/r` is an optional switch for suppressing the actual change of focus.

If the string passed as the first argument matches a name in the Workplace Shell's Window List, then that named task is given the focus by calling the WinSwitchToProgram API function.

The name mask may contain wildcard characters, the asterisk and question mark, which have their usual DOS command line meanings – the asterisk matches zero or more characters, while the question mark matches exactly one character. For example, the mask *ABC* matches a name containing the letters ABC, whereas the mask *A*B*C* matches a name containing the letters A, B, and C, but not necessarily next to each other.

PMSW Sample Application

The PMSW application is used in OS/2 .CMD files that are batch command files or REXX programs. A .CMD file can invoke PMSW to test whether a named task is already active and to switch focus to that task, or it can launch a task and change focus to that task once the task is available. A REXX program can contain logic such that, if the target DOS application is already active when the REXX program begins, the DOS application receives the focus instead of the REXX program starting another DOS application window.

LAN administrators who have LAN-resident applications of all types can use REXX programs to facilitate access to these applications. A REXX program can examine the current LAN environment, check to see if the resources required for a particular application are now connected, and acquire resources automatically if needed. This can be done for all types of applications.

For example, say that at a customer site the cc:Mail** application resides on the LAN. A Mail program folder can be defined on the desktop, and it can invoke the REXX program MAIL.CMD. This program tests to see whether the target cc:Mail application is now active, and switches focus to it if so. If the cc:Mail application is not active, then MAIL.CMD checks to see if the user is now logged onto the LAN (because a LOGON command has been done), and whether the cc:Mail LAN drive has been connected, before issuing a START command for cc:Mail and switching focus to it.

Figure 21 lists the MAIL.CMD program, and Figure 22 lists the DOS batch file used as the indirect launching mechanism for the DOS application.

Setup Strings for Folders: For folders, the System Object Model (SOM) class definition file is the file

Keyname	Value	Description
OPEN	=ICON;	Open the icon view.
	=TREE;	Open tree view.
	=DETAILS;	Open details view.
ICONVIEW	=[s1,s2,...sn];	Specify styles for icon view.
TREEVIEW	=[s1,s2,...sn];	Specify styles for tree view.
DETAILSVIEW	=[s1,s2,...sn];	Specify styles for details view.
*VIEW styles	=FLOWED	Display using regular columns.
	=NONFLOWED	Display using regular grid, not flowed columns.
	=NONGRID	Display using no grid or columns. Icons are spaced irregularly.
	=NORMAL	Use normal-size icons.
	=MINI	Use small icons.
	=LINES	Lines in tree view.
	=NOLINES	No lines in tree view.
BACKGROUND	=filename;	Bitmap for folder background, a filename in \OS2\BITMAP.
ICONFONT	=size.name;	Font for icon names such as 10 point Helvetica.
TREEFONT	=size.name;	Font for tree icon names.
DETAILSFONT	=size.name;	Font for Details view text display.
WORKAREA	=YES;	This folder is a workarea; if it is now open, it will be reopened at the next boot.
	=NO;	Folder is not a workarea.
ICONVIEWPOS	=x1,y1,x2,y2;	Set the initial icon view position and the size of this folder on the desktop. Coordinates are percentages of offset of the whole screen from the lower left hand corner. x1 sets left side position, x2 right, etc.
ICONPOS	=x1,y1;	Position this icon on the desktop.

Figure 12. Keyname/Value Pairs for the WPFolder Class

WPFOLDER.SC. The class hierarchy for the WPFolder class is:

SOMObject
  WPObject
    WPFileSystem
      WPFolder

Users can create folders by dragging a folder template from the Templates container onto the desktop or onto another folder, and then responding to the system prompts for title, etc.

Users can also create folders using REXX SysCreateObject calls. The object setup-string values that can be used for folders are listed below.

See the section "Setup Strings for Objects" for object settings common to both WPFolder and WPProgram.

Figure 12 shows the keyname/value pairs added by the WPFolder class.

Setup Strings for Programs: For programs, the System Object Model (SOM) class definition file is the file WPPGM.SC. The class hierarchy for the WPProgram class is:

SOMObject
  WPObject
    WPAbstract
      WPProgram

The WPProgram class is the program-object class. It provides an object that points at executable programs, and allows the user to run that program by simply double-clicking on the program object. The program can also contain a variety of useful additional parameters, such as the environment for the program and the parameters that are passed to it. An instance of this class can be created as a Workplace object, and is created initially by the system in its template form. It has the title Program and resides in the Templates folder.

Other instances of this class that are initially created by the system include:

- DOS Full-Screen in the Command Prompts folder
- DOS Window in the Command Prompts folder
- OS/2 Full-Screen in the Command Prompts folder
- OS/2 Window in the Command Prompts folder
- Every object in the Games folder
- Some objects in the Information folder
- Every object in the Productivity folder

Figure 13 shows the keyname/value pairs added by the WPProgram class.

See the section "Setup Strings for Objects" for object settings common to both WPFolder and WPProgram.

Setup Strings for Objects: Figure 14 shows the keyname/value pairs supported by the WPObject class.

Keyname	Value	Description
ASSOCFILTER	=filterslist;	Sets the filename filter for files associated with this program. Multiple values are separated by commas.
ASSOCTYPE	=typelist;	Sets the type of files associated with this program. Multiple values are separated by commas.
EXENAME	=filespec;	Name of program file, or fully qualified filespec. This may be "*" for command processors for OS/2, DOS, or WINOS2. PROGTYPE is required when "*" is used.
NOAUTOCLOSE	=YES;	Leaves the window open upon program termination.
	=NO;	Closes the window upon program termination.
PARAMETERS	=params;	Sets the parameters list, which may include substitution strings. The parameters are included as the command line passed to the target program.
PROGTYPE	=PM;	Presentation Manager session.
	=FULLSCREEN;	OS/2 full-screen.
	=WINDOWABLEVIO;	OS/2 window.
	=VDM;	DOS full-screen.
	=WINDOWEDVDM;	DOS window.
	=WIN;	WINOS2 full-screen.
	=WINDOWEDWIN;	WINOS2 window, common session.
	=SEPARATEWIN;	WINOS2 window, separate session.
(OS/2 2.1) (See note 1 below)	=PROG_31_STD;	WINOS2 full screen, Win 3.1 standard mode.
	=PROG_31_STDSEAMLESSVDM;	WINOS2 window, separate VDM, Win 3.1 standard mode.
	=PROG_31_STDSEAMLESSCOMMON;	WINOS2 window, common session, Win 3.1 standard mode.
	=PROG_31_ENH;	WINOS2 full screen, common session, Win 3.1 enhanced mode.
	=PROG_31_ENHSEAMLESSVDM;	WINOS2 window, separate VDM, Win 3.1 enhanced mode.
	=PROG_31_ENHSEAMLESSCOMMON;	WINOS2 window, common session, Win 3.1 enhanced mode.
SET (see note 2 below)	XXX=VVV;	Set environment variable XXX to a value. This is also used to set DOS settings for DOS and WINOS2 programs.
STARTUPDIR (see note 3 below)	=path;	Sets the working directory.

1. "(OS/2 2.1)" in the Keyname column indicates the start of values for PROGTYPE that pertain only to OS/2 2.1.

   The PROGTYPE=prog_31_xxx values pertain only to OS/2 2.1 and later releases of the operating system. Values above the PROG_31_xxx values can be used for all releases OS/2 2.x.

2. SET values require special escape-character handling when they contain special characters such as commas or semicolons. Use the caret (^) character (the upper-case 6 on US keyboards) as an escape immediately before the special character. A good example of this is the SET DOS_VERSION value, which has commas in it: `SET DOS_VERSION=USERPGM.EXE^,10^,10^,4,OTHER.EXE^,2^,3^,4;`

   In the case of DOS_VERSION, all the values will be replaced by the value you enter.

3. STARTUPDIR for drag-and-drop launching of programs is the directory that contains the dropped data file when a value is not specified explicitly.

Figure 13. Keyname/Value Pairs for the WPProgram Class

Keyname	Value	Description
TITLE	=title;	Object's displayed title string.
ICONFILE	=filespec;	Specifies the .ICO file containing the icon bitmap to display.
HELPPANEL	=id;	Default help panel. The id is usually a resource in a .DLL.
HELPLIBRARY	=filespec;	Defines the help library.
TEMPLATE	=YES;	This object is a template that can be used to create similar objects.
	=NO;	This object is not a template.
NODELETE	=YES;	This object cannot be deleted. Setting this property for an object restricts users from dragging it to the shredder.
	=NO;	This object can be deleted.
NOCOPY	=YES;	This object cannot be copied.
	=NO;	This object can be copied.
NOMOVE	=YES;	This object cannot be moved.
	=NO;	This object can be moved.
NOLINK	=YES;	This object cannot be linked (make a shadow from)
	=NO;	This object can be linked.
NOTVISIBLE	=YES;	This object is invisible.
	=NO;	This object is visible.
NOPRINT	=YES;	This object cannot be printed.
	=NO;	This object is printable.
ICONRESOURCE	=id,module;	This defines the id of the icon resource in the module's dynamic link library (DLL).
ICONPOS	=x,y;	Places the object's icon on the desktop or within a folder or container when object's location is not the desktop. x,y is relative to lower left-hand corner and represents percentages of full width and height.
OBJECTID	=<name>;	This is an identity unique within the system that will stay with the object even if it is moved or renamed as to file name or title. The angle brackets are part of the id string.
NORENAME	=YES;	This object cannot be renamed.
	=NO;	This object can be renamed.
NODRAG	=YES;	This object cannot be dragged.
	=NO;	This object can be dragged.
NODROP	=YES;	This object cannot accept dropped objects.
	=NO;	This object can accept dropped objects.
HIDEBUTTON	=YES;	Views of this object will have a hide button.
	=NO;	Views of this object will have a minimize button instead.
MINWIN	=HIDE;	Views of this object will hide when the minimize button is selected.
	=VIEWER;	Views of this object will minimize to the viewer.
	=DESKTOP;	Views of this object will minimize to the desktop. An icon appears on the "landing area" at the lower left of the desktop.
CCVIEW	=YES;	Create a new view of this object when the user selects open, to provide concurrent views.
	=NO;	Switch focus to the open view of this object when the user selects open.
OPEN	=SETTINGS;	Open a settings view when object is created.
	=DEFAULT;	Open the default view when the object is created. This can be used to create and launch WPProgram objects.

Figure 14. Keyname/Value Pairs for the WPObject Class

```
/* REXX Sample: Load REXXUTIL */
/*
   Check to see if all REXX utility functions have been loaded,
   and load the functions if needed.  This needs to be done only
   once, because the functions remain loaded and accessible.
   NOTE: RxFuncQuery returns 0 if function is loaded already.
*/
entry_point    = 'SysLoadFuncs'
function_name  = 'SysLoadFuncs'
module_name    = 'REXXUTIL'
if RxFuncQuery( function_name ) <> 0 then
   do
      call RxFuncAdd function_name, module_name, entry_point
      call SysLoadFuncs
      if RESULT <> '' then
         do
            say   'RxFuncAdd returned '      ||,
                  RESULT                     ||,
                  ' trying to register '     ||,
                  function_name              ||,
                  '. Program cancelled.'
            exit
         end
   end
```

Figure 15. REXX Sample: Load REXXUTIL

```
/* Figure 16.  REXX Sample: Hidden Object */

TESTSTR = '"*OS/2 CMD*"'
'@PMSW' TESTSTR '/r'
if RC = 1 then
   do
      '@PMSW' TESTSTR;'@EXIT'
   end

ClassName   = 'WPProgram';
Title       = 'OS/2 Cmd'
Location    = '<WP_NOWHERE>'
Setup       =,
   'EXENAME=*;'                        ||
   'PROGTYPE=WINDOWABLEVIO;'           ||,
   'MAXIMIZED=YES;'                    ||,
   'CCVIEW=NO;'                        ||,
   'OPEN=DEFAULT;'                     ||,
   'STARUPDIR=C:\;'                    ||,
   'OBJECTID=<' || Title || '>;'       ||,
   ''
call SysCreateObject  Classname,,
                      Title,,
                      Location,,
                      Setup,,
                      'UPDATE'
'@PMSW' TESTSTR
EXIT
```

Figure 16. REXX Sample: Hidden Object

Sample REXX Programs

This section lists all the REXX programs mentioned in this article:

- REXX Sample: Load REXXUTIL (Figure 15)
- REXX Sample: Hidden Object (Figure 16)
- REXX Sample: DOS Desktop Object (Figure 17)
- REXX Sample: MAKEINI Function (Figure 18)
- REXX Sample: Get Boot Drive (Figures 19a and 19b)
- REXX Sample: SysCreateObject (Figure 20)
- REXX Sample: MAIL.CMD for PMSW Program Call (Figure 21)
- DOS Sample: CCMAIL.BAT DOS Batch File (Figure 22)

Figure 15 shows how to load the REXXUTIL function package that is used by all the sample programs. Once the REXXUTIL.DLL has been loaded, it remains available for other REXX programs until it is unloaded. Loading REXXUTIL.DLL is a prerequisite for all the other sample programs.

One method of ensuring that REXXUTIL.DLL is loaded is to create a shadow object in the Startup folder of the REXXLOAD.CMD program, so that the DLL is loaded. Including code in each REXX program to check if the DLL has already been loaded adds a little overhead and avoids errors. In distributing REXX programs, you should include explicit code to test for and load REXXUTIL.DLL, because other users may not have the same initial environment.

Reference Material

The cited references are available in softcopy form as Information Presentation Facility (IPF) .INF files; some may also be available as harcopy manuals. The softcopy versions of manuals are usually

newer and available sooner than printed versions.

OS/2 2.x Developer's Toolkit. This toolkit contains softcopy information about the Application Programming Interface (API) functions for the OS/2 Control Program, Presentation Manager, DOS/Windows, and System Object Model (SOM). It also contains the Presentation Manager Reference. The toolkit is a component of C++ for OS/2, and is also available as a separate product.

OS/2 Procedures Language/2 REXX. This softcopy reference is part of the base OS/2 2.x online documentation. A hardcopy version of this documentation is available as a separate item from IBM, and the hardcopy version contains additional information about parse statements that is not contained in the softcopy.

Presentation Manager and Workplace Shell Redbook, order number GG24-3732, April 1992, IBM Corporation. This Redbook provides guidance and examples for developing applications for PM and WPS. It is Volume 3 in the Technical Compendium of five volumes, GBOF-2254.

Application Development Redbook, GG24-3774, April 1992, IBM Corporation. This Redbook examines the Presentation Manager execution environment, describes the structure and implementation of PM applications, and illustrates PM's facilities for supporting object-oriented techniques. It also discusses the management of workstation-based application development projects. This is Volume 4 in the Technical Compendium of five volumes, GBOF-2254.

OS/2 2.1 REXX Handbook, ISBN 0-442-01734-0, by Hallet German, published by Van Nostrand Rheinhold. Provides essentials of and extensions to REXX programming.

```
/* REXX Sample: DOS Desktop Object */

ClassName   = 'WPProgram'
Title       = 'QBASIC'
Location    = '<WP_DESKTOP>'
Setup =,
   'EXENAME=\OS2\MDOS\qbasic.exe;' ||,
   'PROGTYPE=DOS;'                 ||,
   'MAXIMIZED=YES;'                ||,
   'CCVIEW=NO;'                    ||,
   'OPEN=DEFAULT;'                 ||,
   'STARUPDIR=\OS2\mdos;'          ||,
   'OBJECTID=<' || Title || '>;'   ||,
   ''
call SysCreateObject  Classname,,
                      Title,,
                      Location,,
                      Setup,,
                      'UPDATE'
EXIT
```

Figure 17. REXX Sample: DOS Desktop Object

```
/*  REXX Sample: MAKEINI Function */

/* assumes REXXUTIL is loaded */
ClassName       = 'WPProgram'
Title           = 'New Object'
Location        = '<WP_DESKTOP>'
Semi            = ';'

Setup =,
   'EXENAME=C:\RESULTS.CMD;'          ||,
   'PROGTYPE=WINDOWABLEVIO;'          ||,
   'OPEN=DEFAULT;'                    ||,
   'MAXIMIZED=YES;'                   ||,
    ''

ApplicationName = 'PM_InstallObject'

KeyName   = Title || Semi || ClassName ||,
            Semi  || Location

Call SysIni    'USER',,
               ApplicationName, KeyName, Setup
```

Figure 18. REXX Sample: MAKEINI Function

```
/*  REXX Sample: Get Boot Drive

   This is method 1 of 2. It assumes that REXXUTIL is loaded,
   and that your desktop is on boot drive
*/

/* get the drive:\path\name of desktop */
ApplicationName = 'FolderWorkareaRunningObjects'
call SysIni 'BOTH',,
            ApplicationName,,
            'ALL:',,
            'Objects'
if Objects.0 <> 1 then
   do
      Say   'Unable to locate ' ||,
            ApplicationName     ||,
            ' in system INI file'
      exit
   end

BootDrive = LEFT( Objects.1, 2 )
say 'Boot drive is' BootDrive
return

/* end of method 1 */
```

Figure 19a. REXX Sample: Get Boot Drive

```
/* REXX Sample: Get Boot Drive

   This is method 2 of 2. It assumes that COMPSEC environment
   variable points to \os2 directory on boot drive
*/
BootDrive = left(value('COMSPEC',,'OS2ENVIRONMENT'),2)
say 'Boot drive is' BootDrive
return

/* end of method 2 */
```

Figure 19b. REXX Sample: Get Boot Drive

The REXX Language, ISBN 0-13-780651-5, by M.F. Cowlishaw, published by PTR Prentice Hall. Mike is the father of REXX. This is the definitive REXX book.

REXX Reference Summary Handbook, ISBN 0-9639854-1-8, by Dick Goran, available as IBM publication S246-0078. Contains descriptions of object setup strings for use with SysCreateObject, as well as general REXX programming information.

REXX From Bark to Byte, IBM Redbook GG24-4199. Many coding examples to do desktop customization.

Acknowledgement

Dick Goran, author of REXX Reference Summary Handbook and an OS/2 Advisor, assisted by imposing coding standards on the REXX examples in Figures 15 - 22, and by providing excellent criticism.

Bruce E. Högman *is a senior systems engineer with 30 years of programming experience on IBM systems, an Air Force Vietnam veteran, and former technical instructor at the Computer Sciences School, Quantico, Virginia. He currently works for Electronic Data Systems Corp. at IBM Boca Raton in the Migration Support Group, which assists companies in OS/2 migration and upgrade projects. Mr. Högman can be reached electronically at CompuServe userid 72050,1327 or Internet userid bhogman@vnet.ibm.com.*

```
/*  REXX Sample: SysCreateObject */

/* Create a folder object, and then
   create a program object in that folder
*/
FolderClassName = 'WPFolder'
FolderTitle     = 'MyFolder'
FolderLocation  = 'C:\'
FolderSetup     =,
                  'OBJECTID=<' || FolderTitle || '>;'
                  ''
call SysCreateObject    FolderClassName,,
                        FolderTitle,,
                        FolderLocation,,
                        FolderSetup,,
                        'UPDATE'
if RESULT = 1 then
   do
      ProgramClassName  = 'WPProgram'
      ProgramTitle      = 'MyProgram'
      ProgramLocation   = '<' || FolderTitle || '>'
      ProgramSetup      =,
                        'EXENAME=C:\TOOLS\MYPRG.EXE;'         ||,
                        'OBJECTID=<' || ProgramTitle || '>;' ||,
                        ''
      call SysCreateObject  ProgramClassName,,
                            ProgramTitle,,
                            ProgramLocation,,
                            ProgramSetup,,
                            'UPDATE'
      if RESULT = 1 then
         do
            Say 'Folder '                    ||,
                '"' || FolderTitle || '"'    ||,
                ' and Program '              ||,
                '"' || ProgramTitle '"'      ||,
                ' have been created'
         end
      else
         do
            Say 'Could not create program ' ||,
            '"' || ProgramTitle || '"'
         end
   end
```

Figure 20. REXX Sample: SysCreateObject

```
/*  REXX Sample:  MAIL.CMD for PMSW Program Call */

/* REXX MAIL.CMD uses PMSW.EXE program calls */

entry_point    = 'SysLoadFuncs'
function_name  = 'SysLoadFuncs'
module_name    = 'REXXUTIL'
if RxFuncQuery( function_name ) <> 0 then
   do
      call RxFuncAdd function_name, module_name, entry_point
      call SysLoadFuncs
      if RESULT <> '' then
         do
            say   'RxFuncAdd returned '   ||,
                  RESULT                  ||,
                  ' trying to register '    ||,
                  function_name           ||,
                  '. Program cancelled.'
            exit
         end
   end

'SETLOCAL'
'C:'
'CD \CCMAIL'
CCMAIL = 'CCMAIL'

parse upper arg FirstName LastName UserPassWord

CCMAILServer   = 'ccmail'
MDRIVE.0       = 0
PMSW_Result.0  = 'FOCUS:'
PMSW_Result.1  = 'READY:'
PMSW_Result.2  = 'ERROR:'

/* Appl always connected to M: drive */
call SysFileTree 'M:\CCMAIL\MAIL.*', 'MDRIVE', 'FSO'
if MDRIVE.0 < 1 then
   do
      '@NET USE M: ' CCMAILServer
      Net_RC = RC
   end
else  Net_RC = 0

if Net_RC = 0 then
   do
      bREADY = 0
      'PMSW' ccmail '/R'
      if PMSW_Result.RC \= 'READY:' then
         do
            'START "CCMAIL" /C /MIN /WIN c:\CCMAIL\CCMAIL.BAT',
                  FirstName LastName UserPassword
         end
      else
         do
            'PMSW' ccmail
            if PMSW_Result.RC = 'FOCUS:' then
```

Figure 21. (1 of 2) REXX Sample: MAIL.CMD for PMSW Program Call

```
            do
               bREADY = 1
            end
         end
         do while (bREADY < 1)
            'PMSW' ccmail '/R'
            if PMSW_Result.RC = 'READY:' then
               do
                  'PMSW' ccmail
                  if PMSW_Result.RC = 'FOCUS:' then
                     do
                        bREADY = 1
                     end
               end
            else
               do
                  call SysSleep 2
               end
         end
   end
else
   do
      if Net_RC = 2 then
         do
            'net use m: /d'
         end
      say "UNABLE TO ACCESS CCMAIL"
   end

byby:
'@EXIT'
```

Figure 21. (2 of 2) REXX Sample: MAIL.CMD for PMSW Program Call

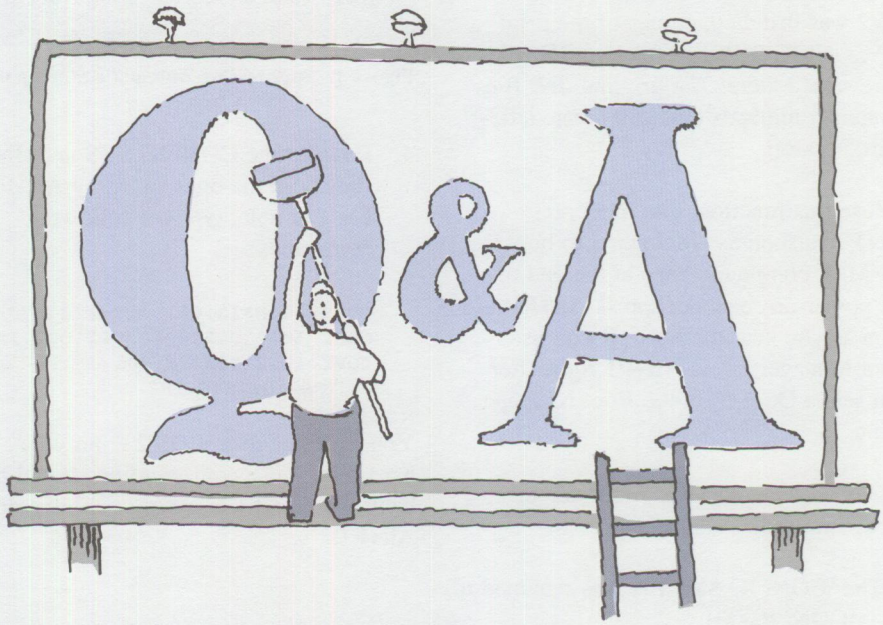
The author's REXX code examples included in this article are available from CompuServe, the IBM OS/2 BBS, and other sources as files or packages named PMSW and RXMP.

```
@ECHO OFF
C:
CD \CCMAIL
CD M:\CCMAIL
SET CCCONFIG=C:\CCMAIL\CCMAIL.INI
SET PATH=M:\CCMAIL;%PATH%
SET TEMP=C:\
SET WP=/D-C:\WP51/PF-C:\WP51/M-VIEW
IF X==X%1 GOTO NONAME
IF X==X%3 GOTO NOPASS
M:MAIL /N"%1 %2" /P"%3"
GOTO EXIT
:NOPASS
M:MAIL /N"%1 %2"
GOTO EXIT
:NONAME
M:MAIL
:EXIT
EXIT
```

Figure 22. DOS Sample: CCMAIL.BAT DOS Batch File

OS/2 Questions and Answers

Doug Azzarito
IBM Corporation
Boca Raton, Florida



Q:

I originally installed OS/2 on a FAT partition, but I'd like to switch to the High Performance File System (HPFS). How can this be done?

A:

While no FAT-to-HPFS conversion utility is available at this time, you can convert a FAT-based OS/2 system to HPFS. You'll need a reliable data backup and restore solution – something that every prudent computer owner should have.

Converting to HPFS involves backing up all your existing files, reformatting the drive as HPFS, then restoring all your files (but there's a trick, so read on). How you back up depends on the backup software you use. Once you have a total backup (don't forget the hidden files), make sure you have the software you need to restore the backup after the disk drive is reformatted.

Boot OS/2 from your OS/2 installation diskettes, and insert OS/2 Diskette 1 when prompted. At the "Welcome to OS/2" screen, press Esc to get to an OS/2 full-screen session. Insert OS/2 Diskette 2, and you're ready to reformat your disk. Assuming C is the drive you are converting, use the command:

```
FORMAT C: /FS:HPFS
```

Once the format is complete, you'll need the trick I mentioned before:

There is a file in the root directory of the OS/2 boot drive called OS2BOOT. This is the OS/2 "Mini File System Driver," or mini-FSD. The mini-FSD file must match the file system of the boot drive. Since your boot device was originally FAT, the OS2BOOT file you backed up is for the FAT file system.

The SYSINSTX command will create the HPFS OS2BOOT file for you. SYSINSTX.COM is usually found on the OS/2 Install diskette. However, in order to work on HPFS, it needs access to UHPFS.DLL, which is on OS/2 Diskette 2. To get these files (SYSINSTX.COM and UHPFS.DLL) together, copy them both to your newly formatted hard disk, and type:

```
SYSINSTX x:
```

where `x:` is the drive you just formatted. This creates the file OS2BOOT in the root directory, but it is marked hidden (`DIR /A` reveals it). Now, when you restore your data, make sure you don't overwrite the OS2BOOT file!

Once the restore is complete, make sure the first line of your CONFIG.SYS file looks something like this (as a single line):

```
IFS=C:\OS2\HPFS.IFS /CACHE:256
    /CRECL:4 /AUTOCHECK:C
```

The `CACHE` parameter specifies the size of the HPFS cache (in KBytes). `CRECL` specifies the maximum size of disk transfers that are cached (in KBytes). The `AUTOCHECK` parameter specifies which drives should be checked when you boot OS/2. Make sure that *all* HPFS drives are listed. Also make sure that the HPFS.IFS file is in the OS/2 directory. If not, copy the file from your installation diskette.

You're now ready to boot OS/2 from your new HPFS boot drive!

Q:

When I upgraded to the OS/2 2.11 ServicePak, the instructions mentioned pressing Alt+F1 to restore the Desktop. What exactly does this do? Are there other "secret" key combinations in OS/2?

A:

The Alt+F1 key combination rebuilds your OS/2 configuration to a "newly installed" condition. When you use Alt+F1, OS/2 replaces your CONFIG.SYS, your OS2.INI and OS2SYS.INI, and your Desktop. This is

done using files stored in the \OS2\INSTALL subdirectory; when OS/2 was installed, copies of the original CONFIG.SYS and the two .INI files were stored there. The original .INI files contain commands that will create a default Desktop.

To use this function, you must press Alt+F1 as soon as OS/2 starts to boot up. Most computers beep at the end of the power-on sequence; press Alt+F1 as soon as you hear the beep. If you use Boot Manager, press Alt+F1 right after you select OS/2 from the Boot Manager menu.

If OS/2 detects Alt+F1, you'll see messages similar to:

The \CONFIG.SYS file was renamed to \CONFIG.001.
The \OS2\OS2.INI file was renamed to \OS2\OS2.001.
The \OS2\OS2SYS.INI file was renamed to \OS2\OS2SYS.001.

These files were restored from copies in the \OS2\INSTALL subdirectory.

Your original Desktop is saved, and a new one is created.

This operation is useful for re-initializing your OS/2 system. However, remember that many OS/2 programs place configuration information in the CONFIG.SYS or .INI files. Therefore, you should only use Alt+F1 when you want to erase any configuration changes made since installation.

If you press Alt+F1, and then decide you want to reverse its effects, do the following:

1. Boot from your OS/2 installation disk
2. Insert disk 1 when prompted
3. At the "Welcome to OS/2" prompt, press Esc to open an OS/2 full-screen session.

```
SET RESTARTOBJECTS=STARTUPFOLDERSONLY
```

Figure 1. Suppressing Automatic Startup of Programs

4. Replace the CONFIG.SYS and .INI files with the ones saved by Alt+F1. For example, type the following commands:

```
COPY \CONFIG.001 \CONFIG.SYS
COPY \OS2\OS2.001 \OS2\OS2.INI
COPY \OS2\OS2SYS.001
   \OS2\OS2SYS.INI
```

You can now restart OS/2 from your hard disk. Your configuration should be back to what it was before you pressed Alt+F1.

Ctrl+Shift+F1 can come in handy if you shut down with lots of programs still running.

If you open the Drives folder and look at your boot drive, you'll see two Desktop folders, but only one of them should have the diagonal hatch-marks, which tell you it is open. The Desktop folder that is not open is the one that Alt+F1 created for you, and it can be deleted. Since Desktop folders are marked "Undeletable" in the Workplace Shell, it's easiest to delete it from the command line. Open the Settings of this Desktop folder, and look on the File page to find out its "real" name (it will usually be "DESKTOP1"). Then go to an OS/2 command prompt, and remove the extra Desktop folder and all its subdirectories. (There is no harm in leaving this extra Desktop on your system, but it wastes disk space.)

Another "secret" key-combination that is sometimes confused with Alt+F1 is the Ctrl+Shift+F1 key combination. Ctrl+Shift+F1 is used to tell the Workplace Shell to suppress the startup of programs that were running the last time you ran OS/2. By default, the Workplace Shell restarts all programs that were running when you last shut down.

Ctrl+Shift+F1 can come in handy if you shut down with lots of programs still running. Rather than wait for all these programs to restart when you next start OS/2, use Ctrl+Shift+F1.

Whereas you must press Alt+F1 as soon as your computer reboots, Ctrl+Shift+F1 is only recognized by the Workplace Shell, so you have to wait until after OS/2 starts the Workplace shell. Once OS/2 switches the screen to graphics mode, press and hold the left Ctrl key, the left Shift key, and the F1 key until the Desktop finishes loading.

If you never want programs to restart automatically, add the line in Figure 1 to your CONFIG.SYS file.

This tells the Workplace Shell to suppress startup of any program, unless you put that program in the Startup folder.

Doug Azzarito *is an advisory programmer on the OS/2 Change Team. He has worked on OS/2 development projects since 1986. Doug is co-author of RBBS-PC, the industry-standard bulletin board software for personal computers. He received a BS in computer science from the University of Florida in 1982.*

The OS/2 Corner: Why Buy OS/2 Now?

Len Zakas
Channel Islands PC Users' Group
Camarillo CA

Reprinted with permission from The Outer Edge, *newsletter of the Channel Islands PC Users' Group, April 1994 issue, and subsequently updated by the author for publication in this magazine.*

The following analysis is provided in case you are still on the fence about OS/2 versus Windows 4 (a.k.a. Chicago).

Why *should* you consider OS/2 2.1 for your personal computer?

Positive Reasons: OS/2 2.1 and OS/2 for Windows are the only PC-based, 32-bit operating systems that are shrink-wrapped and on the shelf at software stores right now. They take full advantage of the 32-bit capabilities of the 386, 486, or Pentium processor you already paid for. DOS uses only 16-bit data paths (and remember that Windows 3.1 is a DOS program).

OS/2 runs DOS programs better than DOS ever can, because OS/2 has no "640K barrier." OS/2 runs Windows programs at least as well as Windows 3.1 does. Thus, over 50,000 existing programs can be used right now – and most will run better than they do now.

If you have Windows 3.1 tweaked up the way you like it, but wish it would not keep giving you those %@?# errors that make you reboot, then OS/2 for Windows is what you need. This is OS/2 2.1 without the Windows 3.1 code; it uses your existing copy of Windows on your machine, but doesn't mess with DOS's limitations.

OS/2 pre-emptively multitasks programs, so that most problems experienced with one program will not automatically require rebooting the computer. In OS/2, a "poorly controlled" Windows or DOS program can usually be closed with Ctrl+Esc and two mouse clicks. The Ctrl+Alt+Del "three-fingered salute" is not a constant necessity.

Separate Virtual DOS Machines – VDMs – can be set up for each program. The result is that you can use all your available RAM to run programs, not just the measly first megabyte that DOS allows.

The OS/2 Workplace Shell that you see on the monitor quickly becomes intuitive, useful, and flexible. (These are the kinds of praises usually reserved for the Macintosh**.) But, if you prefer, OS/2 can also set up your desktop to look just like Windows 3.1.

OS/2 does not need a memory manager; it just assigns whatever memory a program may need, without regard to any 640K barrier. It allocates memory in a much more dynamic way than does DOS, and protects each program that is running from interfering with any other program's chunk of memory.

And OS/2 comes with a print spooler and full multimedia capability, comparable to the Amiga** computer (the industry standard). How much additional money would this cost a DOS or Windows user?

OS/2 will be in its third or fourth (depends on how you count them) significant update by the end of 1994. It is a mature operating system, with almost six million copies sold.

What About Windows? Microsoft has released Windows 3.11. It was designed to be incompatible with the new OS/2 for Windows product (although there is a simple fix). Is this the first blow in response to OS/2's success? More importantly, instead of spending effort to improve a product, Microsoft has chosen to make their product less compatible. We users pay the final price for this type of competitive response.

Microsoft says to wait until at least December 1994 (or is it now sometime in 1995?), when Windows 4.0 (Chicago) will be released. But buried in their press releases is that current DOS and Windows programs will run the same way as they run in Windows 3.1 – no improvement! Only newly purchased (= $$$) applications will be capable of being pre-emptively multitasked – as *all* applications already are with OS/2! Ctrl+Alt+Del will continue to be a Windows "standard."

Additionally, Windows 4.0 will require all-new utilities, because products written for Windows 3.x will not work. You will have to pay $__ (fill in your guess) to upgrade your Norton Utilities** and PC Tools programs. Call your favorite software company now to find out their planned upgrade prices and schedules.

If You Still Believe in Microsoft . . . For those who still believe only in Microsoft, keep waiting for Windows 4.0, then break out the checkbook to buy the very first version of an operating system that will be less capable and years less mature than OS/2 2.1 is now. (Then be ready to buy the upgrade fix that is sure to follow two or three months later.)

In the meantime, IBM will have continued to improve and expand the capabilities of the 32-bit operating system that works now.

***Len Zakas**, who resides in Camarillo, California, is a computer and management instructor. He actively supports the efforts of all the computer user groups in Ventura County. He is vice president of Assisting Children to Excel, a non-profit organization that provides computers and computer training to children from low-income families.*

What I Learned About OS/2: A Testimonial

Ramon Abasolo
Tokyo, Japan

This is a testimonial about OS/2 from a computer user who says he is "not from IBM and not from Microsoft, just an ordinary guy who loves computers."

I used to be a die-hard Windows fan, totally impressed with Microsoft products. Recently, I installed OS/2 in my system, partly because two friends whose opinions I respect had advised me to try it.

Talk about a total perception shift! I am totally impressed at OS/2 and have begun to wonder what the big deal is with Chicago.

Let me set the stage here. I am not a PC expert. Only once have I opened a PC chassis, and I've never installed a hardware device. I read a lot of Windows-related manuals and magazines.

My observations about OS/2:

1. People complain that installing OS/2 is difficult compared to installing Windows. I agree that OS/2 installation is challenging, but I disagree that it is more difficult than installing Windows. I think the reason for this perception is that a lot of PCs come with Windows pre-installed. The process of installing OS/2 is almost the same as installing Windows: you have to know your installed devices, your IRQs and DMAs, your device drivers, and whether the device is compatible. Didn't we spend a lot of time in Windows tinkering with cache sizes, swap-file sizes, driver configurations? It's the same thing in OS/2. I'd bet that if someone re-installed DOS and Windows from scratch, he/she will encounter the same installation problems.

   Actually, I installed OS/2 in 30 minutes. Configuring it to work with an array of video cards, sound cards, CD-ROM drives, and printer was the principal cause of my woes. I attribute it to device incompatibilities – particularly with my cheap, pre-installed, $99, Sound Blaster-wannabe sound card. This is probably the only caveat I see with OS/2 – make sure your hardware is compatible!

2. There is also the misconception that OS/2 is unstable, or that its performance is sluggish. I, too, had that misconception, because my thinking was that "if Microsoft could not do it, then no one else can." Was I wrong! OS/2 was very stable. No GPFs. Very rare system crashes. In fact, my "crashes" were often due to the individual application sessions, and can be attributed to wrong settings.

   Performance? Almost the same as in Windows. Experts say that DOS applications run faster in OS/2, but I haven't noticed it. People think that Windows applications will run slower in OS/2; I haven't noticed that, either. I admit, though, that when I opened multiple sessions, especially if an application is running in the background, the performance tended to deteriorate. A lot of tuning will have to be done for the individual sessions, particularly with the cache size, swap-file size, and memory requirements. But, again, isn't it the same in Windows?

   Speaking of memory, I am running a 486/66 with a 1 MB video card and 8 MB of RAM.

3. Another frequent misconception is the incompatibility, or the total lack, of applications. I admit a lack of 32-bit OS/2 applications, but you can run DOS and Windows programs with almost no problems.

   To all those who use DOS for games, I have news for you. Almost all of my games, including SimCity, X-wing, Tie-Fighter, DOOM, Master of Orion, and X-com, run with no problem! Once I even made a mistake of running a game in an OS/2 windowed session, and it still ran! Dynamix's Front Page Football (a DOS game) requires a lengthy (20-30 minute) simulation of a week's football games – so I simply minimized the session, let it run in the background, fired up my Quicken for Windows** to enter my day's expenses, returned to FPS, and – voila! simulation completed! Try doing *that* in Windows!

People have come to believe that Windows is the only worthwhile product to offer a graphical user interface. But Windows short-changes people who believe that it has "true" multitasking capability. Don't get me wrong – I think Windows is a great product that has steepened the learning curve for all computer novices. Just don't believe all the marketing hype.

Speaking of "hype": now I look at all those articles, ads, opinions, and forecasts about Windows 95 in a new light. Dozens of articles have been saying that Windows 95 will solve all the woes of Windows. But now I say to myself: What's the big deal? OS/2 does that already.

If you want a full 32-bit multitasking system, you can do one of two things – wait for Windows 95 in 1995 (and it will require some time after that to stabilize), or get OS/2 now. My advice? Don't wait – get OS/2 now!

***Ramon Abasolo**, a Filipino citzen, is the Japan Regional EDP Auditor at Union Bank of Switzerland. His prior employment includes EDP Production Control Head at Citibank Philippines, and EDP Security Head at Citibank Japan. Ramon's computer expertise is primarily in mainframe systems and software, and he acquired his own PC a year ago. His Internet userid is 71125.570@compuserve.com.*

IBM OS/2 LAN Server Version 4.0: A Stable, Interoperable, Manageable Performer

Linda Nesmith
IBM Corporation
Austin, Texas

This article covers the highlights of LAN Server 4.0 from the perspective of end users and customers.

While most press attention has been directed at Novell** NetWare servers, IBM has been developing and enhancing its own OS/2 LAN Server. Based on IBM OS/2, the LAN Server runs OS/2, DOS, and Windows clients and applications.

Over the years, IBM OS/2 LAN Server has found its way into many corporate accounts worldwide, where IBM marketing has traditionally been strongest. For example, nearly half of all U.S. banks use LAN Server.

Today, IBM OS/2 LAN Server is a platform that is recognized for its:

- stability
- support of access to multiple servers with a single logon
- interoperability
- resource location transparency to users
- high performance
- manageability

LAN Server 4.0 enhances these characteristics with new ease of installation, ease of use, and ease of administration.

New Graphical User Interface: LAN Server's most noticeable new feature is its new Graphical User Interface (GUI). Over 2000 beta users of LAN Server 4.0 worldwide gave the new GUI very positive early feedback. "The total Graphical User Interface and the online documentation make it possible for anyone to be an administrator," says Robert Labenski of Client Server Networking, a Connecticut-based company.

In fact, most of LAN Server's code has been completely rewritten to eliminate the Microsoft LAN Manager legacy and to focus on making the product easy to use and expand. Particular attention has been paid to installation and administration, including new integrated tuning capabilities for optimizing server performance in your particular network configuration.

Enhanced Peer-to-Peer Capability for OS/2, DOS, and DOS/Windows Clients: Popular features such as the OS/2 Peer-to-Peer capability have also been enhanced, so that DOS and DOS/Windows clients, as well as OS/2 clients, can use them. In fact, much work has been done on DOS and Windows user support in LAN Server 4.0. In addition to peer support, performance has been enhanced, and an enhanced TCP/IP stack has been added (TCP/IP is now a native protocol in the IBM LAN Server product).

OS/2 Stability Extended To Client/Server Environments: LAN Server's long list of standard features include many that exploit the client/server platform and extend it to a multi-system environment. LAN Server was first in the industry with the now-popular concept of *domains*. Domains give you the ability to have a network that might be made up of several cooperating servers which users see as a single-server system, and which can be accessed by as many as 1,000 users (on each individual server in the domain). LAN Server maintains security of the data in a domain by allowing access to be defined to the granularity of each individual user for each served resource (file, print queue, program, or asynchronous communications port).

Interoperability: Interoperability with a variety of servers is an area where LAN Server continues to improve. Exploiting the advantages of OS/2 allows a LAN Server user to access resources on other systems with full server security. Such other systems include the IBM LAN Server versions that offer Macintosh support or run on the AS/400*, RS/6000*, or mainframe hosts (MVS and VM), as well as servers from Novell, Microsoft, SUN, and DEC**.

IBM LAN Server 4.0 even includes a Network SignOn Coordinator utility that allows a user to gain access to all required systems with a single logon. In fact, a LAN Server user at a DOS requester can have full access to all of the aforementioned systems with only a single protocol stack, thus saving valuable RAM in the DOS requester machine.

Protecting Current Investments While Enabling New Directions: IBM has stayed true to its commitment to protect customers' investments by maintaining compatability with previous IBM LAN Server releases – in fact, LAN Server 4.0 still supports requester code written in 1985 for IBM's original DOS Server's requesters! This is clear demonstration of IBM's commitment to the LAN

Server as a key part of its client/server strategy – a base to build on in the future. And LAN Server can grow as a business grows, with no revolutionary conversions required to implement new features (such as the future DCE).

New Easy Install: Functions like those described above are beyond the needs of many users, and attention has been paid to the first-time user of a server, too. Naturally, the first thing a purchaser does with the server code is to install it. An experienced, knowledgeable user can choose to go through an Advanced Install and select many specific tailoring options. But new users can select the Easy Install, answer six quick questions, and be on their way!

Whichever installation method is chosen, the user will find – online – easily searchable help for each step of the procedure. Once installed, the new administrator signs on and sets up the new domain by defining users and groups, other server machines (if any), shared resources, and access permissions.

Migration Path For Legacy and Competitive Servers: Novell and Microsoft users have already found, or soon will find, that they are faced with migrating to new products from those companies. But IBM's new migration utilities make it easy for those users to choose LAN Server instead.

A new utility, free from IBM, even automates this task for new LAN Server users who currently have NetWare 2.2 or 3.1x servers. This migration aid allows a drag-and-drop of definitions from the old NetWare server to the new LAN Server, and does all required conversion tasks automatically, under the covers. If your legacy server is LAN Manager, there is a migration aid for you, too!

Drag-and-Drop Administration: If there is no conversion to automate, the administrator will find using the new GUI to be simplicity itself...

- Define a user, and an icon representing that user appears.
- Define a resource, and an icon representing that resource appears.
- Drag the user (or a group of users) and drop them on a resource's icon, and the users will be given logon access to the resource automatically.
- Have one user defined, and want others to be copies of that definition (logon resource accesses, application menus, etc.)? Just click on "Make another" and enter the new users' IDs – task complete!
- Want to add a new application to the users' menus? Drag its icon to the group's icon, and drop it – task complete!
- Have multiple servers in a domain, and want to move a file from one server to another to do some load balancing? The aptly named MOVESTUFF utility not only copies the data and erases the old copy, it also changes the servers' resource definitions so they point to the new copy at its new home – automatically. The next time users log on, they will transparently be accessing the resource at its new server home.

Plus, all these functions work not only within, but also across, domains of servers.

Experienced users of other servers may have noticed there has been no mention of logon scripts. There are none!

Although logon scripts are supported by the IBM LAN Server, many LAN Server users have never had to write a logon script. Logon scripts normally give users access to resources at servers, but this is done automatically in IBM LAN Server. Logon scripts also display the users' menu, but this too is automatic in IBM LAN Server. Now, logon scripts are needed only when a company wants to add its own functions to the normal IBM LAN Server logon procedure.

IBM OS/2 LAN Server 4.0 – The Best!
If you want to:

– enjoy what is arguably the best server performance in the industry today (the IBM LAN Server is optimized for the Pentium processor)

– enjoy a true object-oriented user interface (the IBM LAN Server is fully SOM2-compliant and integrated with the IBM OS/2 Workplace Shell)

– reap the advantages of a true open client/server platform (because of IBM OS/2, IBM LAN Server does not require a dedicated server machine)

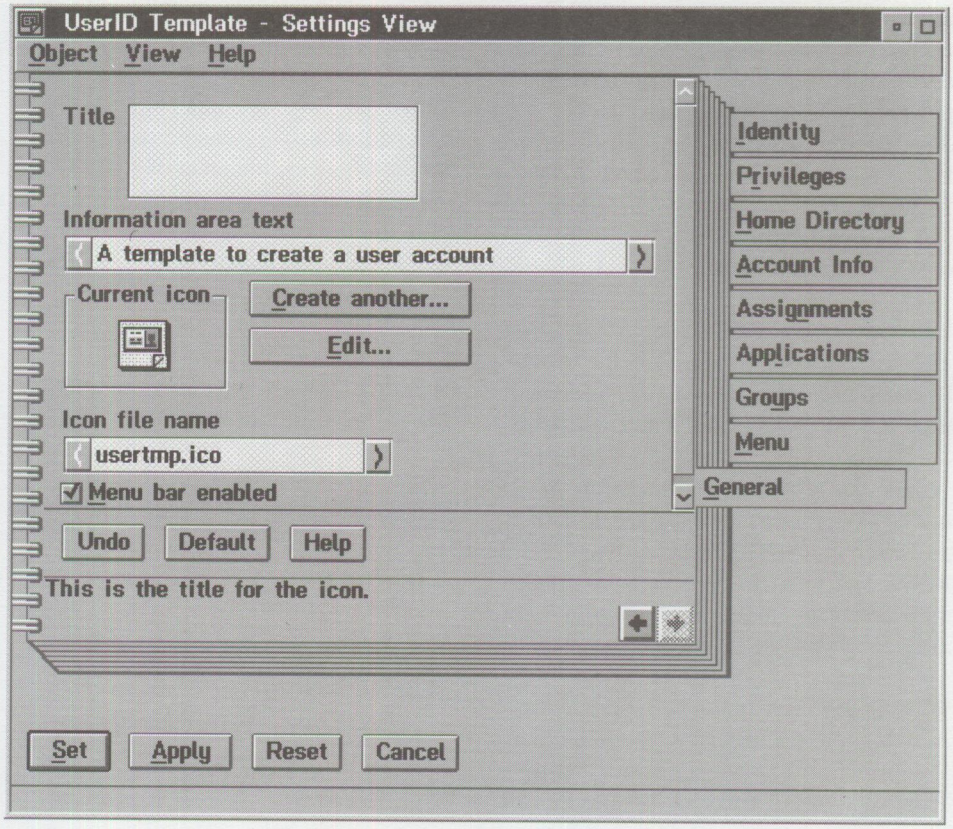
then take a good look at IBM OS/2 LAN Server. With IBM OS/2 LAN Server, you will spend your time running your business, not your servers.

***Linda Nesmith** is in IBM's Personal Software Products division in Austin, Texas and provides support for OS/2 LAN Systems software to IBM's non-U.S. operating units. She has been involved with PC communications software for ten years, both in development and in IBM's European and Asia/Pacific operations. Prior to that, she worked in office-system communications development and large-account marketing. Linda has also also studied relational database technology in graduate school at the University of Texas. She was named to the Marquis Who's Who of American Women for 1991-1992. She can be reached via Internet at nesmith@ausvm1.vnet.ibm.com, and via IBMMAIL at USIB4W88.*

IBM LAN Server 4.0: Networking for the Future

Steve French, Gary Hunt, and Tom Herrick
IBM LAN Systems Development
Austin, Texas

With its release of LAN Server 4.0 in September 1994, IBM has broken new ground. LS 4.0, the fastest, easiest-to-use network operating system that IBM has ever released, is receiving high marks from many computer analysts. This article discusses the design strategy for LAN Server 4.0, highlighting the many enhancements in the product, and giving IBM's future direction for LAN Server. In this article, "OS/2" refers to OS/2 2.1, 2.11, and OS/2 Warp, Version 3.



We in IBM LAN Systems Development began the design of LAN Server 4.0 before LAN Server 3.0 was completed in October 1992. Groups of programmers and designers and usability experts met to discuss ways to improve the product, and every component was analyzed for potential improvement. Many customer requirements were collected, and strongly influenced our product designs. We organized teams (190 people in all) of developers and testers to create LS 4.0. (To see the list of the 190 programmers who worked on LAN Server 4.0, select Help-About from the LAN Server Administration Object.)

LAN Server 4.0 Design Strategy

Our strategy in LAN Server 4.0 was to build a network operating system that improves upon the strengths of LAN Server 3.0, continuing to provide the traditionally strong features – single system image, robust security, industry-leading performance, and high quality – that have made it popular in large corporations.

In addition, LAN Server 4.0 was designed to be much more appealing to small businesses and small departments in larger organizations, by making the product the easiest network operating system to install, use, and administer.

Finally, LAN Server 4.0 was designed to be a first step in the strategy that IBM has defined under the banner of the Open Blueprint. The Open Blueprint, along with its predecessor, the Networking Blueprint, is a structure for a distributed systems environment; it provides the base upon which to build, run, and manage distributed applications using industry standards. LAN Server 4.0 prepares for this environment by providing a transport built on the Multi-Protocol Transport Network (MPTN) architecture, and by changing key protocol flows and structures to prepare for integration of Distributed Computing Environment (DCE) technology in the future.

This overall strategy led to a set of key design goals for the LAN Server 4.0 product to improve its usability, functionality, reliability, and performance, and to make migration easy.

Network Operating System for Large and Small Users: Network Operating Systems give customers the ability to distribute and centralize key enterprise resources, such as data, printers, and I/O devices. This ability is significant to both large and small enterprises. In large or growing installations, the key facility is the ability to add servers without adding complexity. LAN Server has always had the concept of domains, which addresses this requirement well. In a small installation, the key requirement is management of the LAN Server features in the simplest way, because the users are also the administrators of the network.

In previous versions of LAN Server, administrative facilities were based on in-

terface technologies introduced in early versions of OS/2. OS/2 2.0 and subsequent releases have demonstrated the power and ease of use of an object-oriented interface through features like the LAN-aware Workplace Shell. LAN Server 4.0 provides state-of-the-art, object-oriented user interfaces like the ones in the OS/2 Workplace Shell.

LAN Server 4.0 addresses other key usability issues by simplifying the installation and setup process. Large installations are often especially concerned with the cost of repetitive, large-scale administration. LAN Server addresses this in the graphical user interface, and also by providing complete command-line and REXX interfaces that enable an administrator to create batch (.CMD) files to manage the LAN Server environment.

With its new user-interface facilities that are based on leading-edge technology, simplified installation and setup, and reduced cost of repetitive or large-scale administration, LAN Server 4.0 is able to better serve the needs of diverse customer sets with a single product.

Industrial-Strength Platform: LAN Server 4.0 adds a number of new features and functions that improve the LAN Server environment, by allowing administrators to manage and control the environment more effectively, and by allowing users and applications to communicate and share resources in new and better ways. Examples are features like enforced disk-space limits, network-enabled DDE and Clipboard, and peer services for DOS and Windows Clients.

LAN Server 4.0 also supports and exploits high-performance server platforms such as symmetric multiprocessing (SMP) systems and Pentium servers.

LAN Server is being rolled out into larger and more complex installations every day, and customers rely more and more on it as an industrial-strength platform. Each LAN Server release contains fixes to all previous releases, and LAN Server has consistently improved its stability and reliability in each release.

Improved Testing and Problem Determination: To ensure that LAN Server 4.0 meets the level of stability and reliability that our customers require, the LAN Server test labs grew substantially in capacity. Our test labs thoroughly exercised the new product features and ensured no regressions in existing functions. Scenarios were modeled after complex customer environments, and were run seven days a week, 24 hours a day to assure round-the-clock availability and reliability. Also, an integration test lab was added to test the coexistence and interoperability of complex combinations of products, both from IBM and from other vendors.

LAN Server 4.0 was voted "Best of Show" by Data Communication/LAN Times at the NetWorld+Interop trade show.

Problem determination services were improved in LAN Server 4.0 to aid in the location and resolution of problems. Development processes were continually analyzed for improvements to catch defects as early as possible in the development cycle, with the result that overall code quality is much improved.

Interoperability and Coexistence: Upgrade of a LAN Server 3.0 domain to a LAN Server 4.0 Base domain is possible with no significant migration requirements and no significant loss of interoperability. In addition, LS 4.0 improves interoperability and coexistence with, and migration from, other network operating system environments, such as Windows for Workgroups, Windows NT, and LAN Manager. Since LAN Server is based on the industry-standard X/Open PC networking protocol – Server Message Block (SMB) – SMB servers and clients (developed for almost all major operating systems) from many companies continue to interoperate with LS 4.0.

LS 4.0 servers can coexist in a network of LS 2.0 and LS 3.0 servers, and LS 4.0 servers will fully support unchanged LS 2.0 and LS 3.0 clients. All extensions to LS 3.0, such as LAN Server Ultimedia and LAN Server for Macintosh, continue to be supported with LS 4.0, and in some cases LS 4.0 enhances these products.

Accolades: LAN Server 3.0 was acclaimed as the best-performing network operating system in the industry (in LANQuest Labs' "Netware 3.11, LAN Manager 2.1A, LAN Server 2.0, and LAN Server 3.0 Performance Benchmark Comparison" and elsewhere). LAN Server 4.0 maintains this distinction, having been voted "Best of Show" by *Data Communication/LAN Times* at the NetWorld+Interop trade show in September 1994.

Major Enhancements in LAN Server 4.0

Major enhancements in LAN Server 4.0 include:

- New graphical user interfaces
- Enhanced multi-domain support
- Improved server-management facilities
- Enhanced DOS, Windows, and OS/2 Clients
- Enhanced transports for OS/2, DOS, and Windows workstations
- Improved server performance and capacity
- Simplified installation

Each of these improvements is detailed below.

New Graphical User Interfaces

Several new Graphical User Interfaces are introduced in LAN Server 4.0. The new interfaces for OS/2 Clients include:

- LAN Services Administration GUI
- Network extensions to OS/2 Desktop objects
- Improved network application support
- Network DDE and Clipboard

LAN Server 4.0 also includes a new Graphical User Interface for DOS Clients, as well as enhancements to the existing Windows extensions in the LAN Server 3.0 Windows Client support.

LAN Services Administration GUI: The character-based Full-Screen Interface (FSI) in LAN Server 3.0 is replaced in LS 4.0 with a new object-oriented Graphical User Interface designed to be consistent with the Workplace Shell user interface technology introduced in OS/2 2.0. The new LAN Server Administration GUI is based on a design that utilizes objects and object containers. In this design, the items that make up a LAN Server network are implemented as objects that can be dragged and dropped to perform various management tasks. These objects have Settings folders that define the attributes of each object.

The following objects are introduced in the LAN Services Administration facility:

- Domain Container
- Users Container and User Objects
- Groups Container and Group Objects
- Alias Container and Alias Objects (Directory, Printer, and Serial Device types)
- Public Applications Container and Public Application Objects (DOS and OS/2)
- Local Workstation Object
- Defined Servers Container and Server Objects
- Services Container and Service Objects
- RIPL Clients Container and RIPL Client Objects
- DOS RIPL Images Container and DOS RIPL Image Objects

A key attribute of LAN Server technology is the ability to group servers into a single management and administration unit called a *domain.* A key design goal of the GUI was to make it as easy and efficient to manage a domain of many servers as it is to manage an individual server.

The new LAN Server Administration GUI is based on a design that utilizes objects and object containers.

The *object containment* hierarchy in the GUI accomplishes this design goal. For example, the object containers for objects common to all servers in a domain – such as users, groups, and alias resource definitions – are at the top level of the domain container. With this structure, it doesn't matter how many servers are in a domain – the administrative procedures for managing these objects are the same, and management procedures that must be performed on individual server objects are minimized.

Another key design goal of the GUI was to provide facilities that make management procedures as efficient as possible, to reduce the cost of administering LAN Server installations. This design requirement is implemented in key management functions that use drag/drop operations in our new object-oriented user interface implementation. Drag/drop management functions are supported between the following pairs of objects:

- Users and Groups, to add users to groups
- Users and Aliases, to manage access control and add resource assignments
- Groups and Aliases, to add resource assignments to all users in a group
- Users and Applications, to add application assignments
- Groups and Applications, to add application assignments to all users in a group
- Aliases and Applications, to add resource assignments to applications

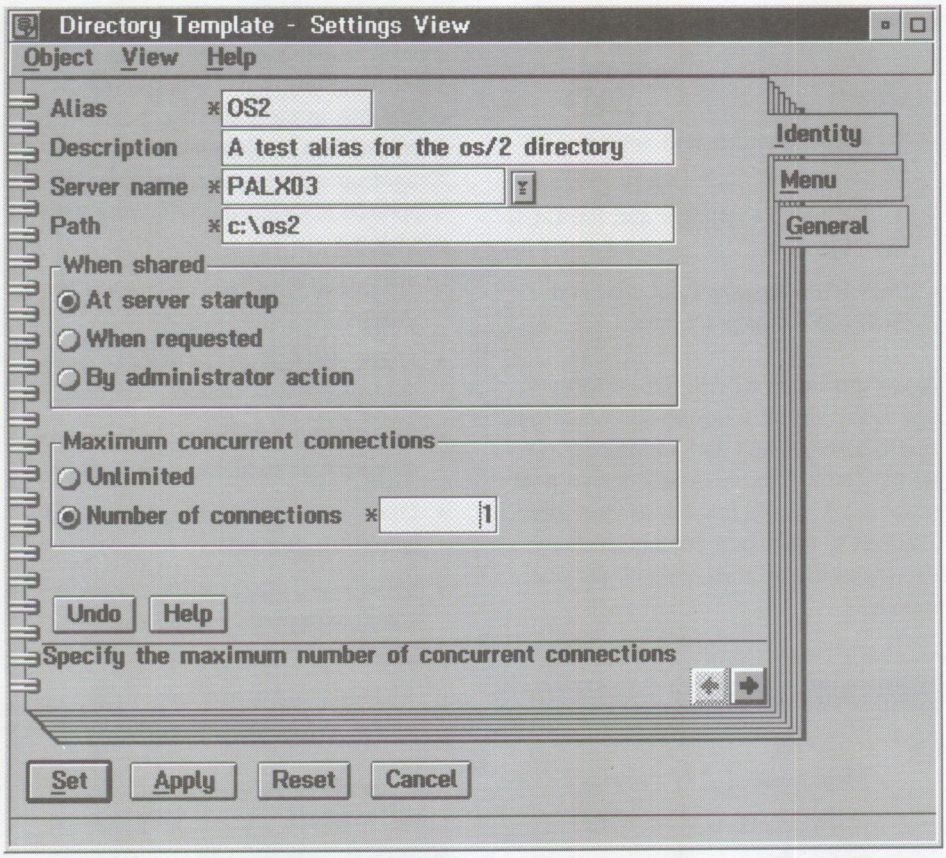
There is also a *shadowed servers* container, to which you can drag defined servers, so that you can create a container of servers that you manage most often.

Another major enhancement of the new interface is the ability to manage multiple domains at the same time. With this capability, LAN Server 4.0 can give administrators a global view of multi-domain installations.

Other enhancements that make the administrator more efficient include:

- User and Group cloning
- Management of all user attributes from a single object
- Configurable templates for all object creations

Two new utilities are added to assist administrators in viewing a server's error and audit logs. To improve messaging, the Netpopup service in LAN Server 3.0 is replaced by a Presentation Manager application that provides an E-mail-like front end to simple messaging. It re-



places the blue VIO pop-up in LAN Server 3.0 with a PM pop-up and dynamic icon to notify the user of new messages.

Two new Graphical User Interfaces are provided for the OS/2 Client, and for the DOS Client running Windows, to support the new Network DDE and Clipboard function in LAN Server 4.0. On the OS/2 Client the interface is implemented using Presentation Manager facilities, and on the Windows Client the interface is implemented as a standard windows application. Both interfaces provide easy-to-use access to the management and configuration of Network DDE and Clipboard.

The LAN Server 3.0 DLR EZVU-based Full-Screen Interface (FSI), which did not support DBCS and did not provide for mouse input, is replaced by a graphical user interface that supports DBCS and mouse input. This GUI can be used on DOS Clients with or without Windows. Under Windows, DOS Client tasks continue to be integrated with the Windows services.

Several enhancements are made to the network extensions that the LAN Server DOS Client provides to the Windows environment:

- *Configurable browse function.* The LAN Server Windows extensions provide the ability to connect to network resources from the Windows File Manager using the LAN Server alias facilities. LAN Server 4.0 extensions are enhanced to allow the user to browse either by domain, using aliases, or by server, using Universal Naming Convention (UNC)-style names. In the case where the alias name space is not available, the browse function automatically provides UNC support.
- *DOS LAN Services management application.* In LAN Server 3.0, client-configuration functions were only available within the Windows Setup application. In LAN Server 4.0, these functions are in a separate DOS LAN Services application. This application supports all the functions available previously, including logon, change password, and send message, as well as new functions, including DOS peer administration (resource sharing) and managing logon assignments.
- *DOS LAN Services Program Group.* A new Windows group collects many network functions into one location. These functions include logon and logoff applications, the new DOS LAN Services management application (WINPOPUP), DOS LAN Services setup, and the new GUI for managing Network DDE and Clipboard.
- *Send Message Application enhancements.* The send-message application is enhanced in LAN Server 4.0 to allow easier viewing and managing of the message log.

Network Extensions to the OS/2 Desktop Objects: Since a set of administrative tasks are best associated with certain OS/2 Desktop objects, LAN Server 4.0 extends the native functions of certain OS/2 objects with network management capabilities. For example, the OS/2 Drive and Folder objects are extended so that the "Start Sharing," "Stop Sharing," "Manage Access Control," and "Manage Size" (disk-space limits) options are available in the context menu for these objects (activated by clicking mouse button 2 on an object). The OS/2 Printer object is also extended with the appropriate set of management functions.

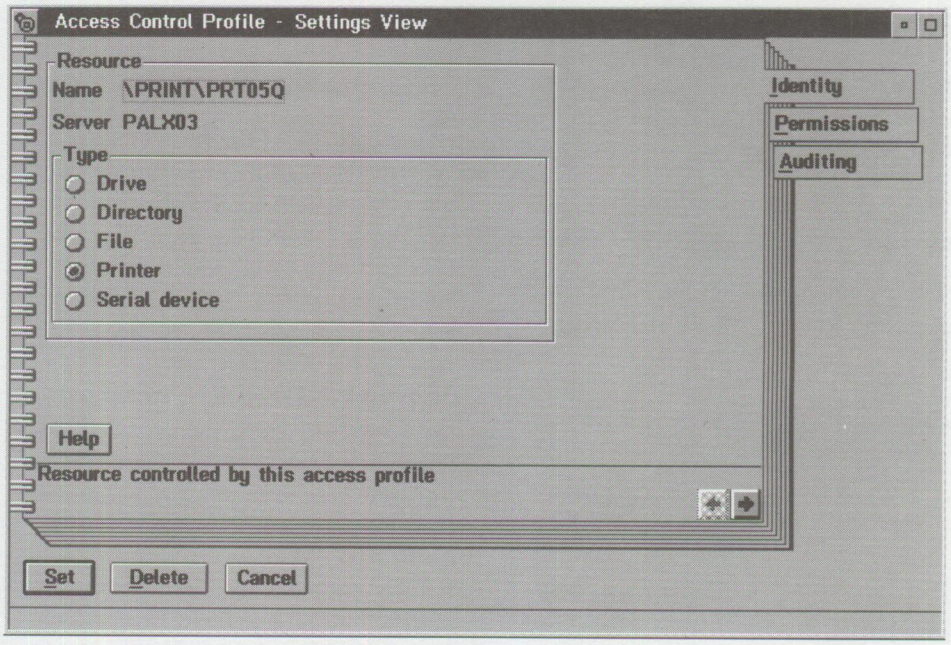
Improved Network Application Support: In LS 4.0, public and private applications have been completely redesigned to integrate with the OS/2 object-oriented Workplace Shell. At logon time, a Network Applications folder is

created on the Desktop. This folder includes the defined public, private, and DOS applications that have been assigned to the user. The program objects that are created for these applications are based on the OS/2 program object, which means that the real icon for the application is displayed, and that settings appropriate to the type of application – including DOS and Windows settings – are available.

Enhanced Multi-Domain Support

As LAN Server customers roll out larger installations, there is a growing need for facilities that are enabled for multi-domain environments. LAN Server 4.0 addresses these requirements with the following enhancements:

- *Cross-Domain Aliases.* Previously, external aliases could be created that pointed to resources outside a user's home domain. However, these external aliases had certain limitations, such as not being available from the command line's NET USE command. LAN Server 4.0 enables a regular alias to point to any server on the network, both inside and outside the domain. These new aliases are called Cross-Domain Aliases, yet they have all the capabilities of normal intra-domain aliases.
- *Multi-Domain Administration.* Previously, a user could have administrator privilege in several domains, but could manage only one domain at a time, because certain administrative functions were available in only the current logon domain. The LAN Server Administration GUI addresses this limitation by allowing up to six domains to be administered from a single logon session.
- *Network SignOn Coordinator.* To assist users with the task of managing their identity on multiple domains, the Network SignOn Coordinator software product is included. This application lets a user create password-change scripts, where one command can be used to change your passwords on several domains at the same time. Network SignOn Coordinator can also support password-change scripts and logon scripts for different sub-systems, making it useful in multi-subsystem environments as well.
- *Multi-Domain Browse and Connect.* Both the OS/2 Workplace Shell and the LAN Server command line's NET USE command are enhanced to enable users to list and connect to aliases in multiple domains from a single logon session.



Improved Server-Management Facilities

Enforced Disk-Space Limits (Advanced Server): One of the most requested customer requirements is the ability to protect a server from users who are not careful with the amount of data they store, or from bugs in applications that cause an infinite loop of data writes to a server until the server runs out of disk space.

LAN Server 4.0 Advanced Server allows limits to be set on the amount of space stored in a directory. The new limits are stored within the file system, and can be applied to any HPFS386 directory. A limit is enforced by returning an error to the user when the limit is exceeded. Also, alerts can be generated as a user approaches disk-space limit thresholds. Facilities are also provided to tune the alert frequency to avoid flooding the network with disk-limit alerts. Improved problem determination and additional tracing and error logging are also added to the Advanced Server.

New Problem Determination Tools: More than 40 new programs are supplied on the productivity-aids diskettes. New and improved problem determination tools include:

- Additional transport error reporting in LANTRAN.LOG
- Duplicate NetBIOS name utility (FINDNAME)
- Redirector buffer dump utility (RDRDEBUG)
- Protocol trace and formatting utilities
- Advanced Server trace and dump facilities
- LAN Server error correlator
- Error log and audit log view utilities

RIPL Configuration Enhancements: Several changes simplify and add to the flexibility of RIPL configuration. API and command-line support for RIPL configuration are examples.

There are also new and updated utilities for managing remote IPL. They include:

- RPLADD - create RIPL definitions
- RPLDEL - delete RIPL definitions
- RPLENUM - list RIPL definitions
- RPLSRID - list RIPL server record identifiers
- RPLDISK - enable or disable local IPL

Enhanced DOS, Windows, and OS/2 Clients

Peer Services for DOS Clients: In LAN Server 4.0, DOS workstations can share their resources with other LAN Server DOS Clients, OS/2 LAN Requesters, and other SMB-based network products. Peer services for DOS Clients have the same single-session restriction that the OS/2 Client has. An unrestricted DOS peer is expected to be offered as a separate product.

Reduced Memory Requirements for DOS Clients: Because DOS real-mode memory is limited, it is critical to minimize using it. This minimizing is achieved both through ongoing optimization efforts and the addition of the Virtual Redirector implementation in the DOS Client when running with Windows. The new DOS Client supports a new lightweight DOS transport called LAN Support Program Lite. All these factors combine to make over 600 KB of real memory available to applications on the new LAN Server DOS Client.

Improved DOS Client: The DOS Client is significantly re-worked to provide better structure and cleaner implementation of key client functions. New functions are added, such as client-side caching and an expanded NET command. The new DOS Client in LAN Server 4.0 is renamed from DOS LAN Requester to DOS LAN Services.

CID Enablement for DOS Client: With the availability of DOS Configuration, Installation, and Distribution (CID), our DOS Client code must be CID-enabled as well, to allow for attended, lightly attended, and unattended installation from a code server.

Network DDE and Clipboard

Network DDE and Clipboard allow application programs or users to copy text from one workstation to another through the Clipboard. You can also create dynamic data links between different workstations. Dynamic data links are copies of data that are automatically updated across the network when changes are made to the original data.

To use Network DDE and Clipboard, you must use applications that support the Clipboard and dynamic data-linking functions. You can use Network DDE and Clipboard with clients running either Windows or OS/2; this function is not available on clients running DOS.

Extended Command-Line Interface: Before LAN Server 4.0, certain functions could be accomplished only through the full-screen interface, and could not be performed through the command-line interface. LAN Server 4.0 includes command-line interfaces for all tasks that were formerly achievable only through the full-screen interface, including common administrative tasks like defining applications, assigning logons, applying access control, and printing the domain definition. These additions enable these tasks to be performed within command files, and to be performed remotely using the NET ADMIN command.

Extended UPM Character Set and Lengths: LAN Server 3.0 has limits for the lengths of user names, passwords, domain names, and computer names. These limits are increased in LAN Server 4.0 to give customers more flexibility in their naming, and to improve interoperability and migration with Microsoft LAN Manager-based products like LM 2.2, Windows for Workgroups, and Windows NT.

LAN Server Enterprise Migration and Interoperability: Several changes are made in LAN Server 4.0 Base in anticipation of the changes required for the enterprise add-on features to LAN Server 4.0 that integrates LAN Server into Distributed Computing Environment installations. These changes encompass customer migration and interoperability issues, to allow servers to be more easily migrated between the environments, and to allow unchanged LAN Server 4.0 Base clients and servers to participate more effectively in the Enterprise environment. Certain tactical changes are made to ease our coding efforts in LAN Server 4.0 Enterprise, and to get a head start on the changes that are required.

Enhanced APIs and Toolkit: LAN Server APIs are enhanced to provide better 32-bit support, and to extend the functionality of some APIs. In addition, instead of simply shipping headers and libraries, we are including the softcopy *Programmer's Guide and Reference* and sample programs in a selectively installable component called the LAN Server Development Toolkit. The *Programmer's Guide and Reference* is also included as part of the softcopy books installation.

Enhanced Transports for OS/2, DOS, and Windows Workstations

The OS/2 transport services in LAN Server 4.0 introduce a transport-independent programming framework based on the Multi-Protocol Transport Services (MPTS) architecture for distributed OS/2 applications. This transport supports sockets programs written to the BSD 4.3 sockets interface standard, and enables them to communicate using any protocol implemented to the transport

framework. LAN Server 4.0 includes the TCP/IP and NetBIOS protocol stacks implemented in this fashion. The OS/2 transport also includes access to NetBIOS through traditional NetBIOS programming interfaces.

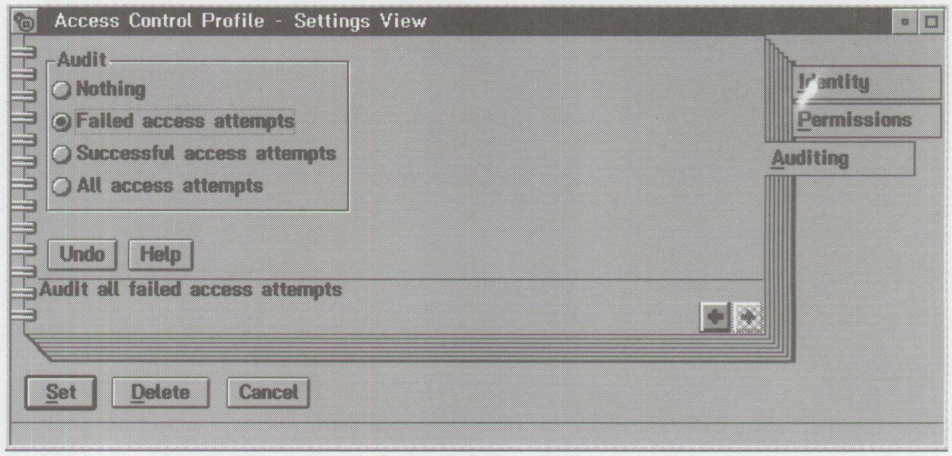
Additionally, the OS/2 transport in LAN Server 4.0 provides an improved implementation of NetBIOS for TCP/IP called TCPBEUI. Previously, the TCP/IP protocol was accessible from NetBIOS applications, but the implementation was inefficient because of transitions through the code paths between the kernel level and the application level. TCPBEUI provides a kernel-level NetBIOS programming interface that accesses the TCP/IP protocol stack at the same kernel level. With this implementation, TCPBEUI offers the most efficient possible NetBIOS access to TCP/IP.

Both the OS/2 and DOS/Windows transports also support a wider range of Network Interface Cards (NICs), which are automatically identified at installation time, thus eliminating the need for the user to specify the type of NIC installed.

Improved Server Performance and Capacity

In previous LAN Server releases, we increased the number of clients that LAN Server can support, by allowing for multiple network adapters and by enhancing the performance of the server. Unfortunately, server buffers could become a bottleneck. LAN Server 4.0 increased the maximum number of server buffers, and dramatically and more efficiently uses memory above 16 MB if available. The key improvements are detailed below.

Hardware Exploitation: LAN Server 4.0 Entry and Advanced packages support symmetric multiprocessing (SMP) Intel machines running OS/2 for SMP. SMP compatibility is especially useful for application serving. Since LAN Server already makes very efficient use of CPU resources and is usually constrained instead by disk I/O capacity, users will typically see dramatic performance benefits with SMP only when running CPU-intensive applications (like databases or transaction processing) on the server, rather than when simply file-serving.



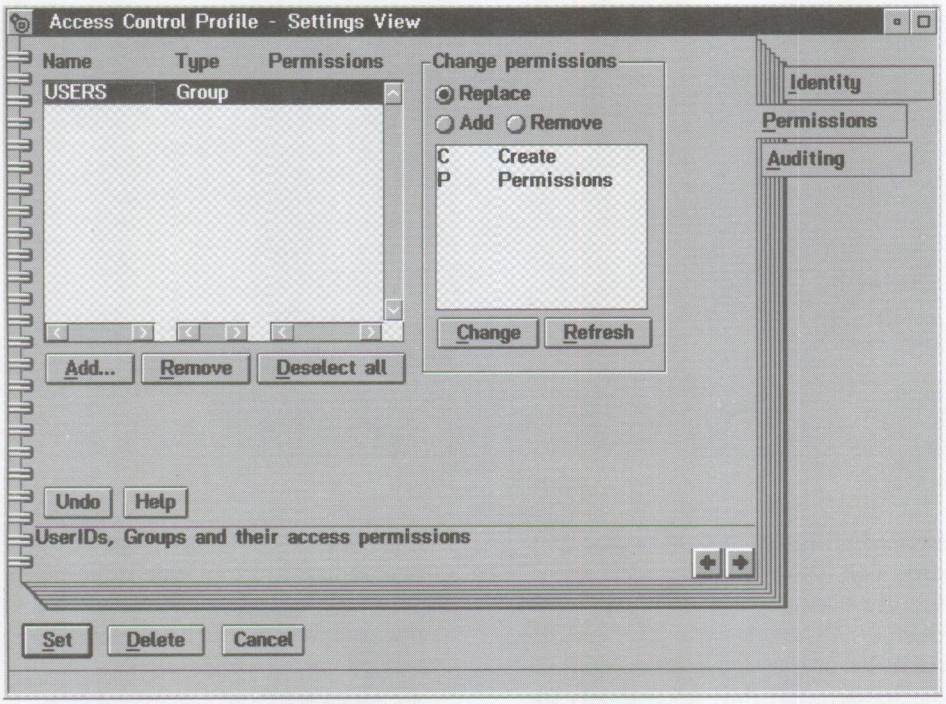
The Advanced Server is also optimized to take advantage of unique features of the Pentium chip – the Advanced Server's performance improves when it determines that the Pentium is present.

LAN Server Ultimedia Advanced Server: The changes previously made in the Advanced Server are carried into LAN Server 4.0. LAN Server 4.0 Advanced Server (with its unique enhancements) can be installed over LAN Server Ultimedia without losing the LAN Server Ultimedia features. This brings to LAN Server 4.0 the improvements made for LAN Server Ultimedia in the Advanced Server. Note that the Resource Reservation System (RRS) of LAN Server Ultimedia is still required to provide LAN Server Ultimedia function; this function is not provided in LAN Server 4.0.

Simplified Installation

LAN Server 4.0 is enhanced with the following features:

- "Easy" install path. A new "Easy" installation process is introduced in LAN Server 4.0. Its design goal was to eliminate as many installation-time questions as possible. When you use this path, many installation parameters get defaults, and questions that are asked are mostly answered by a yes/no choice.
- Automatic detection and identification of LAN adapter. Both the OS/2 and DOS installation programs are enhanced to automatically detect and identify the type of LAN adapter installed on your computer. If the adapter is one of the supported adapters (that is, an adapter whose device driver is shipped with LAN Server 4.0), the user is told which adapter is identified, and that the appropriate device driver will be installed.
- Integration of LAPS and LAN Server Install. In LAN Server 4.0, the installation process for the OS/2 transport (LAPS) is combined with the LAN Server installation process, so that both components can be installed in a single installation step.
- MAKEDISK utility integration. In LAN Server 4.0 Advanced, the installation procedure includes an option for making an HPFS386 boot diskette, which eliminates needing to run the MAKEDISK utility (or to create an HPFS386 boot diskette).
- A comprehensive LAN Server Tuning utility is added.



LAN Server Future Directions

LAN Server 4.0 represents a significant step forward in the LAN Server family of products, and it reinforces IBM's commitment to the LAN Server product line. LAN Server's future direction revolves around several key strategies:

- *Providing LAN Server technology on key platforms*, for example:
  - LAN Server for AIX
  - LAN Server/400
  - LAN Server for MVS/VM (LFS/ESA)
  - PowerPC
- *Integration of DCE technology and implementation of IBM's Open Blueprint.* LAN Server's current directory and security services will be migrated to DCE's Cell Directory Service and Registry Service, providing single system image across an enterprise of heterogeneous systems. The use of open, industry-standard APIs and services will ensure that customers can integrate hardware and software from IBM and other industry providers into a seamless system. DCE will bring to LAN Server new levels of scalability, a more portable platform for developing distributed applications, more advanced security options, and more flexible, global naming services.
- *Protocol independence.* LAN Server's ties to NetBIOS will be broken, and LAN Server will write to the Sockets API on top of the MPTS framework. This will allow LAN Server to be easily adapted to run over any number of protocols, including TCP/IP, NetBIOS, SNA, IPX, and others.

The combination of an extensive LAN Server product family, the right services, and a strong strategy for the future ensures that LAN Server will continue to meet customer needs and solve customer problems for many years to come.

Steve French *is a staff programmer in LAN Systems within the IBM Personal Software Products division in Austin, Texas. Since joining IBM in 1989, he has worked on LAN Server as a designer and (currently) as the development lead for the LAN Requester. He was also responsible for the design of the SMP enhancements to LAN Server. Steve has published many articles and has patent applications pending for his work on LAN Server. He has a BA degree in Computer Science and a Master's degree in Electrical Engineering, both from Rice University. His Internet userid is stevef@ausvm1.vnet.ibm.com.*

Gary Hunt *is an advisory programmer in LAN Systems, IBM Personal Software Products division, Austin, Texas. He is currently the design lead for LAN Server 4.0. Gary has worked on all LAN Server releases in various roles as development manager, technical planner, and designer. He joined IBM in 1975 and has an MS degree in Electrical Engineering from the University of Texas. His Internet userid is gthunt@ausvm1.vnet.ibm.com.*

Tom Herrick *is a staff programmer in LAN Systems, IBM Personal Software Products division, Austin, Texas. He is currently the technical lead in design and development of future versions of LAN Server. After getting his BS degree in Computer Science from The Georgia Institute of Technology in 1989, Tom joined IBM and the LAN Server team. Since then, he has developed and supported all LAN Server versions since 1.0. His Internet userid is herrick@ausvm1.vnet.ibm.com.*

OS/2 LAN Server Entry 4.0 – Highlights

- Server software runs on OS/2 2.1 workstations; a dedicated server is not required for small workgroup networks
- Supports a broad range of client workstations, including OS/2, DOS, DOS/Windows, and others
- Simple to use, with drag-and-drop Graphical User Interface (GUI)
- Easily installed and set up
- Provides peer-to-peer sharing
- Presents users and administrators with a single information resource image, even if there are many physical servers
- Entry-level Network Operating System (NOS) that lets you add users and servers incrementally, as you need them
- Specialized LAN administration skills are not required
- Supports up to 80 users comfortably
- Can be readily extended into other platforms, including AIX*, AS/400, VM, and MVS
- Supports most popular topologies, including Ethernet and Token-Ring
- Supports TCP/IP and NetBIOS protocols

OS/2 LAN Server Entry 4.0 – Details	
LAN adapters supported	Ethernet Token-Ring PC Network LAN adapter cards with NDIS-compliant device drivers
LAN topologies supported	Ethernet Token-Ring PC Network Broadband and Baseband Network Device Interface Specification (NDIS) media Fiber Distributed Data Interface (FDDI) Asynchronous Transfer Mode (ATM) (1995)
LAN protocols supported	NetBIOS TCP/IP 802.2 is provided as an additional protocol
Number of users supported	Up to 254 NetBIOS sessions per adapter, with up to four adapters per server; maximum number of attached workstations depends on LAN adapters, wiring characteristics, and workstation resources
Network functions	Printer, file, and serially attached device sharing Application sharing Messaging Network DDE and clipboard Remote IPL (DOS and OS/2) Multiple network servers User logon/password protection Resource administration Peer-to-peer file and print
Ease-of-use facilities	Graphical online help and an installation tool Drag-and-drop administration Integrated performance tuning tools Graphical OS/2 clients

OS/2 LAN Server Entry 4.0 – Details (continued)		
Software requirements	Server	OS/2 2.1 or higher
	OS/2 Client	OS/2 2.1 or higher
	Windows Client	Windows 3.1 or higher
	DOS Client	DOS 3.3, 5.0, 6.x or higher
	PC LAN Program (PCLP) Client	PCLP 1.34 (Base Service)
	LAN Server 4.0 will interoperate with existing LAN Manager, Windows for Workgroups, and Windows NT clients	
Systems supported	Any personal computer with Intel i386, i486, Pentium, or SMP microprocessor (or compatible)	
Memory requirements (excluding operating system, user application, and user data requirements) (exact requirements depend on which functions and LAN protocol you select)	Server	3MB minimum; an additional 5MB is required when using the GUI for LAN administration
	OS/2 Client	2.5MB minimum; an additional 5MB is required when using the GUI for LAN administration
	Medialess OS/2 RIPL Client	12MB
	DOS Client	Total memory required: 146KB-203KB Below 640KB (low memory): as little as 30KB (depending on your system configuration and assuming use of memory management techniques)
Disk space requirements	Server	18MB
	OS/2 Client	16MB
	DOS Client	4MB (14MB when TCP/IP for DOS is the LAN protocol)

OS/2 LAN Server 4.0 Advanced – Highlights

- Provides industry-leading performance together with high-availability features
- IBM's stragetic Network Operating System (NOS) that lets you add users and servers incrementally, as you need them
- Supports up to 1000 users simultaneously connected to a single server
- Connects a virtually unlimited number of users in a multiple-server domain
- Supports a broad range of client workstations, including OS/2, DOS, DOS/Windows, and others
- Interoperates with existing LAN Server, Microsoft Windows for Workgroups, and Windows NT clients
- Provides peer-to-peer file and print sharing
- Organizes multiple servers and information resources into a single image (domain)
- Allows you to add servers to the domain without paying additional fees for existing users
- Server software runs on OS/2 2.1 and provides browse, connect, and administration services to multiple domains from a single point of control
- Can be readily extended onto other platforms, including AIX, AS/400, VM, and MVS environments
- Simple to use, with drag-and-drop, object-oriented Graphical User Interface (GUI)
- Exploits Pentium architecture for maximum Pentium and LAN Server Advanced performance
- Supports advanced hardware environments with Symmetric Multiprocessing capabilities
- Supports most popular topologies, including Ethernet, Token-Ring, and FDDI
- Supports TCP/IP and NetBIOS protocols

OS/2 LAN Server 4.0 Advanced – Details	
LAN adapters supported	Ethernet Token-Ring PC Network LAN adapter cards with NDIS-compliant device drivers
LAN topologies supported	Ethernet Token-Ring PC Network Broadband and Baseband Network Device Interface Specification (NDIS) media Fiber Distributed Data Interface (FDDI) Asynchronous Transfer Mode (ATM) (1995)
LAN protocols required	NetBIOS TCP/IP 802.2 is provided as an additional protocol
Number of users supported	Up to 1000 simultaneous connections per server
Network functions	Printer, file, and serially attached device sharing Application sharing Messaging Network DDE and clipboard Remote IPL (DOS and OS/2) Multiple network servers User logon/password protection Resource administration Peer-to-peer file and print

OS/2 LAN Server 4.0 Advanced – Details (continued)		
Ease-of-use facilities	Graphical online help and an installation tool Drag-and-drop administration Integrated performance tuning tools Graphical OS/2 clients	
Software requirements	Server	OS/2 2.1 or higher
	OS/2 Client	OS/2 2.1 or higher
	Windows Client	Windows 3.1 or higher
	DOS Client	DOS 3.3, 5.0, 6.x or higher
	PC LAN Program (PCLP) Client	PCLP 1.34 (Base Service)
	LAN Server 4.0 will interoperate with existing LAN Manager, Windows for Workgroups, and Windows NT clients	
Systems supported	Any personal computer with Intel i386, i486, Pentium or SMP microprocessor (or compatible)	
Memory requirements (excluding operating system, user application, and user data requirements) (exact requirements depend on which functions and LAN protocol you select)	Server	5MB minimum (excluding HPFS 386 cache memory requirements); an additional 5MB is required when using the GUI for LAN administration
	OS/2 Client	2.5MB minimum; an additional 5MB is required when using the GUI for LAN administration
	Medialess OS/2 RIPL Client	12MB
	DOS Client	Total memory required: 146KB-203KB Below 640KB (low memory): as little as 30KB (depending on your system configuration and assuming use of memory management techniques)
Disk Space Requirements (excluding operating system, user application, and user data requirements)	Server	19MB
	OS/2 Client	16MB
	DOS Client	4MB (14MB when TCP/IP for DOS is the LAN protocol)

LAN Server Interoperability with Microsoft Windows for Workgroups

Charlie Brown
IBM Corporation
Austin, Texas

IBM LAN Server (LS) and Microsoft Windows for Workgroups (WfW) are generally compatible and interoperable when communicating with each other across a local-area network. They use the same transport protocol (NetBIOS) and the same file-sharing protocol (SMB, or Server Message Block).

This article describes in detail the environments you may encounter when attempting to connect LS and WfW in a network. It also attempts to help you understand how these environments work, and gives you some helpful hints and tips acquired through experience in our test labs and with customers.

Two basic environments are described in this article:
(1) WfW clients accessing shared LS resources on either a full server or an LS requester with peer services;
(2) LS requesters accessing shared WfW resources.

Not covered in this article are installing DOS LAN Requester on top of WfW, and running Windows networking under OS/2 in Windows emulation mode (WIN-OS/2).

This article covers the current versions of software – WfW version 3.1 and 3.11, and LS version 3.0 and 4.0. If a specific version is not mentioned, you should assume that the information is valid for all current versions.

IBM LAN Server (LS) and Microsoft Windows for Workgroups (WfW) both use standard, open protocols for communication. NetBIOS, an IBM invention and *de facto* standard, is used for network- and session-level protocols. Server Message Block (SMB), an X/Open standard, is used for application-level protocols. Because of this use of standards, LS and WfW can and do interoperate successfully.

Starting with LAN Server 4.0, IBM specifies that LS and WfW will interoperate, and includes WfW interoperability in its test suite. Many customers have also successfully hooked LS and WfW together. For example, Jerry Pournelle, author of the "Chaos Manor" column in *BYTE Magazine*, wrote in the May 1994 issue about his positive experiences with integrating LAN Server and Windows for Workgroups on his personal Ethernet LAN.

In general, WfW clients receive equivalent function from LS servers compared to WfW servers. And, WfW clients can take advantage of the improved security, performance, and reliability of LS servers as compared to WfW servers.

However, WfW clients cannot use some of the additional function that LS provides for its own requesters, such as single-system image (aliases), network applications, and logon assignments.

On the other hand, OS/2 LS requesters accessing WfW servers see significant functional degradation compared to LS servers. For example, Workplace Shell operations are severely curtailed; security is less stringent; and long filenames and extended attributes are not supported. DOS LAN requesters accessing WfW servers should see no functional degradation.

Because of these functional deficiencies of WfW servers, we recommend that data to be shared between OS/2 LS and WfW requesters be stored on an LS server, except for casual or occasional use.

Transports: IBM's and Microsoft's implementations of NetBIOS transports are compatible with each other, and communicate with each other across a network. You must, of course, install the correct LAN adapter drivers for the LAN adapters on each computer. LS and WfW support different, but overlapping, sets of LAN adapters. Drivers are not interchangeable between LS and WfW.

IBM and Microsoft both have implementations of NetBIOS over TCP/IP. This article does not cover NetBIOS interoperability over TCP/IP implementations, although some LS users have been successful connecting the IBM and Microsoft implementations of NetBIOS over TCP/IP.

WfW as a Client to LAN Server

When using WfW as a client to LAN Server, several aspects must be taken into account.

Logon: WfW supports both local (workgroup) logon and domain logon. The default is workgroup logon. If you want to have your userid and password validated by an LS domain, you should set up for domain logon. To do this under WfW 3.11, open the control panel's Network icon, and find the settings for Enterprise Logon under Startup Options. Enter the domain name and other information requested.

If you do not log on to the domain, you can still use LS resources, as long as your userid and password are valid for the resource you are connecting to. Home directories, logon assignments, network applications, and PROFILE.BAT are not supported from WfW clients, because these features are unique to LS.

Security: WfW clients are subject to the same authentication and access-control restrictions as LS requesters. Your userid and password must be valid

in the LS domain, and the LS administrator must have granted access to your resource for your userid. Access control is enforced whether or not you have logged onto the domain.

Browsing Shared Resources, Servers, and Domains: WfW clients support browsing of LS domains, if at least one WfW server is present in each domain. To place a WfW server in an LS domain, simply assign it a workgroup name equal to the domain's name. The WfW server in the LS domain provides browsing services to WfW clients, making itself and the other LS servers in the domain visible to all WfW clients. If you have only a few WfW clients accessing a LAN Server domain, the easiest approach is to put the WfW clients into the domain by assigning them a workgroup name equal to the LS domain name, and to have at least one of the WfW clients set up as a server. This way, all LS and WfW resources are visible to all WfW clients.

To access a domain that does not have a WfW server in it, you must know the name of the server you want to access. Type the server name into the path field of your network connect dialog, preceded by two backslashes. Then press Enter. WfW then browses the shared resources on that server.

WfW does not support the LS concept of single-system image and alias names for shared resources. You must know, or use browsing to discover, the server name and network name of shared resources before you can connect to them.

Using File Resources: WfW clients can use LS file resources, including CD-ROM drives, as long as the files and directories have names that are compatible with DOS and Windows. In other words, the file and directory names must be in the 8.3 form, NAME.EXT, where NAME has 1 to 8 characters, and EXT has zero to 3 characters. Files and directories whose names are not compatible with DOS and Windows will not be visible to WfW clients.

Network Operating System	Domain (Workgroup)	Computer	Share (Netname)	Alias	User	Password
LS 3.0	8	8	8	8	8	8
LS 4.0	15	15	15	8	20	14
WfW	15	15	15	N/A	20	8

Figure 1. Name-Length Limits for LS Requesters when Accessing WfW Peer Servers

Using Print Resources: WfW can use LS print resources with no known restrictions. The WfW Print Manager can be used to view and manipulate LS print queues (subject, of course, to security checks).

Sending and Receiving Messages: WfW 3.1 had no capability to send and receive LAN Messages. WfW 3.11 adds a Message Popup utility that is interoperable with LS messaging. If you start the Message Popup utility on WfW 3.11, you will see LS messages sent to you, including print job completion messages. WfW 3.11 can also send messages to an LS requester or server.

WfW Network Applications: The network applications Network Mail, Network Clipboard, Network DDE, Chat, and Network Hearts are provided in WfW. They do not interoperate with LAN Server, although they can continue to be used between WfW systems. LAN Server provides an implementation of Network Clipboard and DDE for both OS/2 and Windows, but this is not interoperable with the WfW Network Clipboard and DDE.

Using Serial Device Resources: WfW cannot share or use serial-device resources with LS.

Using Named Pipes: WfW can act as a named-pipes client to an LS OS/2 server or to OS/2 Peer Services requesters. Named-pipes distributed applications can be implemented between WfW and LS, as long as the WfW application is the client.

LS Requesters Accessing WfW Peer Servers

This section details considerations for LS requesters that access WfW servers and their shared resources.

Naming: If you are using a LAN Server 3.0 requester, you should be aware that not all WfW names are supported. For example, LS 3.0 user names are normally limited to eight characters, while LAN Server 4.0 supports all valid WfW names. Figure 1 details the limits on name lengths.

In LS 3.0, some subsets of function support longer names than described in Figure 1. In particular, the command-line interface (`NET` command) frees you from many of the length restrictions.

It is easy, in WfW, to define domain names and share names with imbedded blanks. To access these names with the command-line interface (`NET` command), use double quotes (") around the string containing the name. For example:

```
NET USE X: "\\my wfw
   server\my share"
```

Logon: WfW servers do not support domain logon. Your LS requester must log on to an LS domain, or must log on locally. LS 3.0 DOS requesters cannot log on locally, so they must log on to a domain. LS 4.0 DOS requesters, as well as LS 3.0 and LS 4.0 OS/2 requesters, support local logon.

To log on locally with an OS/2 requester, you can use the `/V` parameter. For example:

```
LOGON MYUSER /P:MYPASS /V:NONE
```

This command logs you on to your LS requester locally as user `MYUSER`, with password `MYPASS`, and with no validation.

Security: WfW servers do not use userids, only passwords. When you share a directory in WfW, you can optionally specify two passwords: read-only and full (read/write). When your LS requester attempts to access a shared WfW directory, your logon password is compared against the shared directory's read-only and read/write passwords (if any). If there is a match, read-only or read/write access is granted to that directory and to all its files and subdirectories. If there is no match, access is denied. If no password is specified for the shared directory, access is allowed.

If you use the `NET USE` command, you can optionally specify the password for connection to the shared resource, instead of using your logon password. For example, the following `NET USE` command connects to the `RESOURCE` shared directory on the server called `WFWSERV`, whose password is `secret`:

```
NET USE X: \\WFWSERV\RESOURCE
   secret
```

Printer security is handled in a similar fashion.

Browsing Domains and Servers: OS/2 LS requesters can browse WfW domains (workgroups) and servers, if the WfW servers are set up with a special option. This option is specified by adding a line to the SYSTEM.INI file of each WfW server you wish to browse, then rebooting the server. The line appears in the [networks] section, and it reads:

```
lmannounce = yes
```

Once this option is in effect, WfW servers appear to OS/2 LS requesters like additional servers in a domain that has the same name as their workgroup. If this is the same as the LS requester's logon domain, the servers will be automatically browseable by LS requesters.

If the WfW servers are in a different domain, you can still browse them from an OS/2 LS requester. Specify the domain as an "other domain" that you browse. You can do this temporarily with the `NET CONFIG REQUESTER` command, as follows:

```
NET CONFIG REQ
   /OTH:wfwdom1,wfwdom2
```

This command browses two WfW workgroups, `wfwdom1` and `wfwdom2`.

To permanently specify the other domains to be browsed, edit the IBMLAN.INI file on the requester, and add these names to the `othdomain=` line. Up to four other domains can be browsed.

DOS LS requesters at the LS 3.0 level can browse WfW servers that are in the requester's logon domain, and that have the `lmannounce=` option set as described above. DOS LS requesters at the LS 4.0 level can browse WfW servers that are in the requester's logon domain, with or without the `lmannounce=` option.

Using File Resources: LS requesters can use WfW file resources (shared directories), with the following restrictions:

- WfW does not support long filenames or extended attributes.
- WfW does not support 32-bit OS/2 applications that search the directory to find filenames.

Using Print Resources: LS requesters can print to WfW shared printers with no known problems.

Workplace Shell Considerations: The OS/2 Workplace Shell is network-enabled. The Network folder allows you to graphically browse through the network's servers and resources. The Network Printer object lets you view remote printers and drag-and-drop print remotely. The Drives folder sees network-connected drives and lets you browse through their directories and files.

However, because of WfW limitations, most of the Workplace Shell's network enablement does not work with resources on WfW servers. You cannot create a Network Printer object that points to a WfW shared printer. Instead, use the `NET USE LPTn:` command to connect to the shared printer, and define a local printer object for `LPTn:`.

You cannot display WfW shared directories and files through the Drives folder. Use the command line, or a 16-bit application, to view shared directories and files.

The Network folder shows the WfW servers and shared resources, if the `lmannounce=` option has been specified as described above, but below that, the directories and files are not visible. Also, the Network folder does not allow specification of an alternate password for shared resources, so WfW resources protected under a password that is different from your logon password are not visible in the Network folder.

Charlie Brown *is a senior programmer in LAN Systems Development within the IBM Personal Software Products division in Austin, Texas. He is currently the technical lead for Workplace Server development. Since joining IBM in 1977, Charlie has been lead designer of VM/SP, manager of OS/2 SAA, and manager and technical lead for LAN Server development. He holds a B.S. degree in mathematics from Boston University. His Internet userid is browncs@ausvm1.vnet.ibm.com.*

El CID: King of Configuration, Installation, and Distribution

Theodore Shrader and Bob Blair
IBM Corporation
Austin, Texas

Whether a network consists of a handful of scattered workstations or a sea teeming with different makes and models, this kingdom needs a king. The network administrator typically acts as king, but as the kingdom grows, the burdens upon the administrator grow too. Among other tasks, administrators need to be able to quickly and easily install and configure applications across a network. The task of adding new products, new versions, and application fixes to workstations on a network can quickly besiege administrators, and drain their resources of time and energy. By implementing the CID strategy and utilizing CID-enabled applications, administrators can prevent rebellions in the domain, and make themselves and the populace happy.

This article addresses the elements of CID, and how CID-enabled applications can help you, the network administrator, slay the installation and configuration complexity beast.

CID stands for Configuration, Installation, and Distribution. In cooperation with other vendors, IBM has advanced the CID strategy to allow applications to be installed and configured easily via a standard fashion across the workstations in a network environment. (Networked environments include LANs and WANs.) If an application's installation and configuration program is CID-enabled, administrators will know what to expect, and can easily place the application into their software distribution environment.

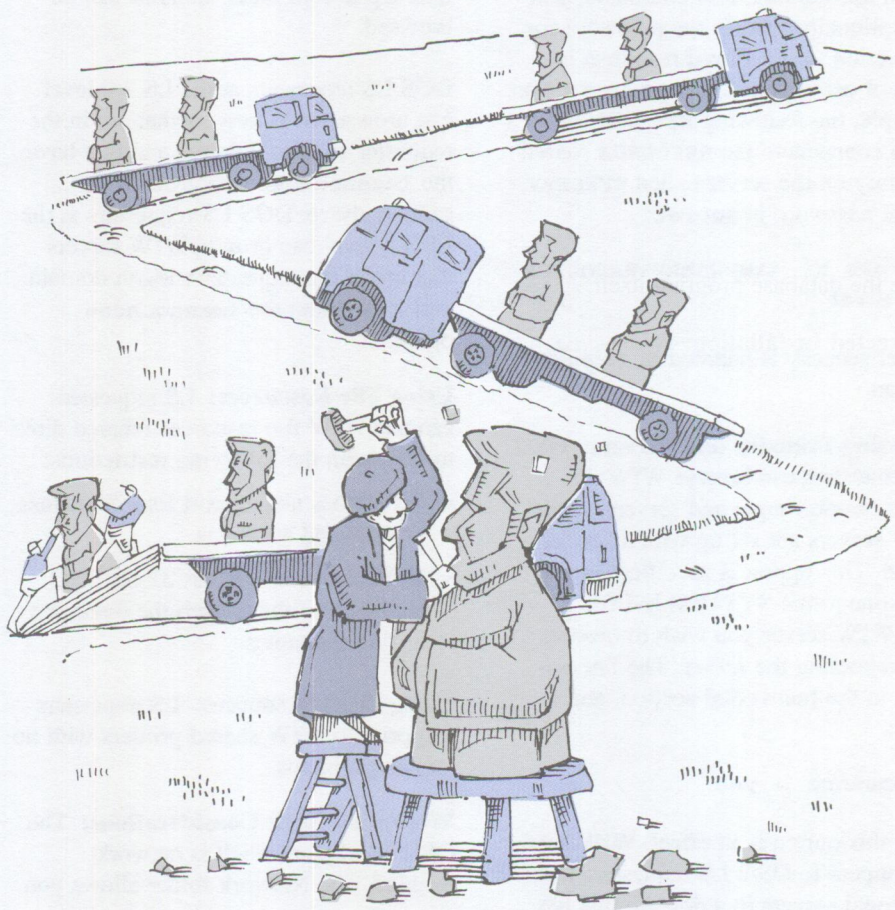
CID-Enabled Applications

A CID-enabled application is one that supports response files in place of user dialogs, and supports redirected installation and configuration from a remote server workstation. These applications have a few other characteristics, too. The application's installation and configuration program must provide CID communication status within the standard set of predefined CID communication codes. This communication status can take the form of return codes, for example. The program must also be able to support a variety of command-line options. However, response file and redirected installation support are the two main features of a CID-enabled application. Both will save you time and expense. Here is why.

Response Files: *Response files* are regular ASCII files composed of keyword-equal-value pairs. These assignments specify installation and configuration options. For example, the following assignment might specify that the database administrator's tools package should be installed on a workstation with a database name of Antietam:

```
DBATOOLS=YES
DBNAME=ANTIETAM
```

By supporting response files, CID-enabled applications do not require users to sit at their workstations and answer panel options and queries. These answers can be predetermined by the administrator before the product is installed or configured. This allows application features to be applied uniformly to multiple workstations. For



example, the administrator could specify that the database client feature should be installed on a group of workstations by creating nearly identical database response files for each workstation. The response files would be different for some keyword and value pairs, because unique values would need to be assigned for the database workstation name. For the most part, each application on each workstation has a specific response file that references global information in a group response file.

In today's world, if the administrator sent out a note listing the options that should be selected, a user installing the database application might accidentally select the database server option on an installation panel, or accidentally type in the wrong database workstation name.

Response files also remove the need for the user to become an expert on the application's installation and configuration options. The database administrator could assign values to these options (keywords in the response file) much more quickly and accurately than users on the network. General users may know how to add rows to tables in the database, but not how to install or configure the database program itself.

Redirected Installation: *Redirected installation* refers to the ability of an application's installation and configuration program to retrieve a product's image files from anywhere on a network. This feature also includes *redirected configuration.* Users who are not connected to a network would normally install programs by patiently jamming disks into their workstations. Once workstations are connected to a network, users can dismiss this time-consuming task, and retrieve their product images directly from a code server that is connected to a network.

Administrators can designate servers on a network as code servers. A *code server* stores images of applications, such as the OS/2 and LAN Server products. It can also store other files, such as log files, which record installation and configuration errors of a product, as well as a history log of the installations and configurations.

Administrators typically create image files for a software product by copying them off the floppy disks containing the application. With the proliferation of CD-ROMs, administrators need not transfer the product images from floppy disks to a hard drive. If a product can be installed from a CD-ROM, administrators can keep the application on the CD-ROM, and users can access the images directly off the CD-ROM on the code server to install and configure applications on their workstations.

Response files remove the need for the user to become an expert on the application's installation and configuration options.

Among other features, the CASSETUP program is specifically designed to help administrators set up code servers easily. CASSETUP is part of the LAN CID Utility set of applets that is shipped with LAN Server 4.0. The "CASSETUP: An Ease-of-Use Tool for LAN CID Utility" article describes the features of CASSETUP in more detail.

Software Distribution Mechanism

CID-enabled applications provide one prong of the CID strategy. The software distribution mechanism provides the second prong to complete the envelopment of the CID problem.

There are three different routes to distribute software:

1. *Attended.* Users manually answer installation and configuration questions, and use product disks to install applications.
2. *Lightly attended.* Users must initiate the installation and configuration sequence and perhaps answer a question or two, but after this initial nursing, users can walk away until the procedure completes. With this route, products are "pulled" from the code server onto the workstation.
3. *Unattended.* Users need not initiate the installation or configuration process. The installation and configuration of applications can take place at night, after the user has left, or when the user logs onto the network. Although this path can initiate a "pull" of product images from the code server to the workstation, this route is characterized by having the code server "push" the product images from the code server to the workstation. A push allows more centralized management and feedback of the installation and configuration process.

Without the CID strategy, administrators and users are stuck with attended installations. CID opens the door to the second and third routes. The third route – unattended installations – provides a more complete management environment than lightly attended installations. Typically, an agent program needs to run on the code server and client workstations to schedule the installation and configuration of products on the network. Once it becomes established in the network environment, administrators and users prefer the unattended route, since it requires the least amount of manual intervention.

Not surprisingly, the lightly attended route is very popular and widely implemented, because it can be done with little maintenance and overhead. Admin-

istrators can use CID-facilitating programs to generate the response files for an application that is targeted for a workstation. These programs can also create special LAN CID Utility (LCU) command files, which help order the installation and configuration of one or more applications on a particular workstation. For example, an LCU command file can install OS/2 on a workstation before installing the Distributed Computing Environment (DCE). It can even take care of workstation reboots between application installations.

With just one disk for DOS and Windows workstations and two disks for OS/2 workstations, users can engage in lightly attended installation. For OS/2, each user needs an LCU command file and a small network redirection program, such as SRVIFS, that allows the user to connect his/her workstation to the network without the overhead of a full-function (unattended) CID client program. Both the LCU command file and thin redirection program can reside on two disks. After booting with the first and inserting the second, users can initiate a lightly attended CID installation and configuration of their applications, and walk away for tea or to pay homage to the network administrator. Typically, the response files for the applications to be installed and configured can reside on a code server that SRVIFS connects to. LCU command files can also reside on the code server instead of on the boot disks.

SRVIFS is shipped with Multi-Protocol Transport Services (MPTS), part of the LAN Server 4.0 product. Add CASSETUP, which is also part of LAN Server 4.0, and administrators can immediately enter the CID realm.

Benefits of Using CID-Enabled Applications

A user on a workstation not connected to a network is an island. These users can still gain some benefit by using CID-enabled applications, such as uniform and predictable command-line options, but the benefits won't be as dramatic as with workstations on a network. Users with standalone workstations can still take advantage of response files, but they won't be able to exploit redirected installation from a code server.

With just one disk for DOS and Windows workstations and two disks for OS/2 workstations, users can engage in lightly attended installation.

Administrators and users of workstations connected to a network gain the most benefit from implementing the CID strategy. By using CID-enabled applications, users need not be knowledgeable about the installation and configuration options of an application. These applications will save them from having to feed product disks into their workstations. Although this is a great way to exercise (the king needs to be fit!) and meet people, administrators need to spend their time on other tasks, not traveling from workstation to workstation installing software and answering questions.

CID – The Strategy for LAN Administrators

Clearly, as more and more companies downsize from mainframes, and as small businesses upsize to LANs, there needs to be a uniform way in which applications can be configured, installed, and distributed. Administrators also face increasing pressures in this task as applications become more complex and new versions and upgrades are shipped. Configuring and installing products and product fixes are not the only tasks that administrators need to deal with. By implementing the CID strategy on their networks, and by using CID-enabled applications, administrators can have a long and fruitful reign.

Theodore Shrader *is a senior associate programmer in IBM Personal Systems in Austin, Texas. He began working for IBM in 1989, and his work has included designing and developing graphical, object-oriented interfaces to database and LAN products. He is also co-author of the* OS/2 2.1 Application Programmer's Guide, *published by Van Nostrand Reinhold in 1994 (ISBN 0-442-01736-7 or IBM # SR28-5449). This book includes a sample CID installation program written in C. Ted holds a B.S. degree in computer science from the University of Texas at El Paso. (Viva Miners!) He can be reached via Internet at shrader@vnet.ibm.com.*

Bob Blair *is an advisory programmer in LAN Systems within the IBM Personal Software Products division in Austin, Texas. He is currently a designer in the area of network software management, and he has had numerous OS/2- and AIX-related design and programming assignments.*

CASSETUP: A LAN CID Utility Ease-of-Use Tool

Bob Blair and Theodore Shrader
IBM Corporation
Austin, Texas

CASSETUP is a tool that automates some of the tasks associated with managing software over a network. CASSETUP is part of the LAN CID Utility (LCU), which is included with IBM LAN Server 4.0. By automating and providing a graphical framework for setup and repetitive steps, such as boot diskette and command file creation, CASSETUP makes the LAN CID Utility significantly easier to use in managing software on a network. This article gives an overview of CASSETUP, followed by an example of its usage.

OS/2 is found more and more in networked environments, where individual workstations are linked together by Token Ring or Ethernet. There may be dozens or even hundreds of OS/2 workstations in such a network.

The presence of a network provides the opportunity to manage various aspects of the workstations' operations from a single workstation. Because of its universal nature and complexity, the management of software installation, configuration, and maintenance is a popular choice for this kind of centralization.

With its LAN Server 4.0 family of products, IBM provides tools for performing several centralized software management tasks. These tools are collectively known as the LAN CID Utility. CID is IBM's strategy for the Configuration, Installation, and Distribution of software in networks. (The article "El CID: King of Configuration, Installation, and Distribution" discusses the benefits of the CID strategy.) Because the tasks associated with software management over a network are still complex, the LAN CID Utility includes a tool that automates some of these tasks. The tool is called CASSETUP.

What is CASSETUP?

CASSETUP is a tool for automating and guiding a user through some of the more complex, time-consuming tasks of software management in a network. It provides a CUA-compliant graphical interface that leads the user through preparation steps for tasks such as installing or upgrading software on remote workstations.

CASSETUP is tucked away in the LAN Server 4.0 package. It is found on the third MPTS (Multi-Protocol Transport Services) diskette, in the APPLETS subdirectory. On CD-ROM, it is in the directory IBMLSE\IBM400N3\APPLETS (Entry Server), or IBMLSA\IBM400N3\APPLETS (Advanced Server).

CASSETUP Makes LCU Easy to Use: CASSETUP adds to LCU's ease of use by:

- Installing and setting up LCU
- Automating routine tasks
- Allowing new IBM and non-IBM applications to be easily managed in the LAN CID Utility framework.

Installing and Setting Up LCU

CASSETUP, once it is installed itself (a process described in the README.UTL file), guides the user through the steps required to install the LAN CID Utility and to set up a CID Code Server. CASSETUP's flexible step-by-step interface makes it easy to set up a Code Server,

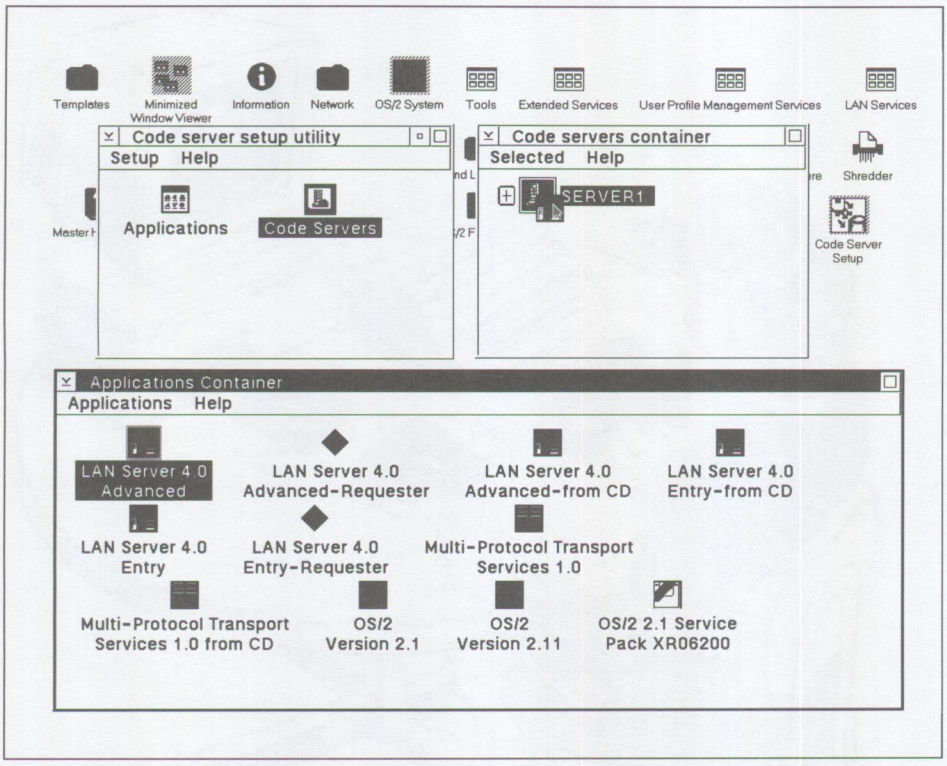


Figure 1. Dropping Code Images on a Code Server

which is a cumbersome and difficult task if done manually.

Automating Routine Tasks: Once the LAN CID Utility is installed and a CID Code Server has been set up, CASSETUP automates the tasks required to:

- Load installable code images on the CID Code Server
- Manage the Code Server
- Build boot diskettes
- Create LAN CID Utility command files

Loading Installable Code Images: To manage software over a network, it is usually necessary to set up a network-accessible copy of the software. A CID Code Server provides the mechanism for sharing installable software on the network, and CASSETUP provides a simple drag-and-drop interface for adding installable software to the Code Server.

Managing the Code Server: The CID Code Server makes installable code available on the network by providing aliases or symbolic names for the actual drive and directory where the code is stored. For example, a Code Server might provide an alias called CID that refers to a physical directory `d:\cidstuff`, under which installable code might be stored. CASSETUP automates the processes of creating new aliases and of associating physical directories with network alias names. As will be seen later, CASSETUP also automatically generates the commands that allow client workstations to use these aliases.

Building Boot Diskettes: Many LAN CID Utility software management activities start with the "two-diskette boot" of the target workstation. A user boots his/her machine from the diskettes, and LCU automatically handles the rest of the installation without user involvement. The software that must be on the two boot diskettes varies according to the programs to be installed. CASSETUP automates the creation of boot diskettes, and provides a flexible way to specify new combinations of software to be put on the boot diskettes.

Creating LCU Command Files: LAN CID Utility command files specify which software management activities will be done on a target workstation. The command files are actually REXX programs that are executed on the managed workstation. They can be created manually using any text editor, and a template command file is provided with the LAN CID Utility, but programming skills are required to customize and extend the template command file.

CASSETUP allows a user to select a list of installable software, and to generate a command file specifying the installation of that set of software. CASSETUP does this in two steps. It first creates an intermediate file called a CASPREP file. This file is much easier to edit than the LAN CID Utility command file, and changing it does not require programming skills. An administrator can make changes to this file, although none should be required in most cases. CASSETUP then creates the command file from the CASPREP file.

Adding New Software: CASSETUP is flexible in the kinds of installable software it can handle. Even third-party and user-developed software can be managed by CASSETUP. Each installable application is defined in an application profile, and all CASSETUP needs to know about a new application is in the profile. The profiles are ASCII files that can be created with any text editor. The profiles included with CASSETUP provide good examples of how profiles are constructed, and the CASSETUP online documentation contains extensive instructions about how to create new application profiles.

CASSETUP Usage Example

Our LAN Software Administrator, Helen, will walk through some sample uses of CASSETUP.

Installing CASSETUP: Helen locates the README.UTL file on MPTS diskette 3 of the LAN Server 4.0 package. She follows the instructions in the README to unzip file CASSETUP.ZIP into a convenient directory, and to run the CASINST command to make CASSETUP ready to use. The CASINST command creates a CASSETUP icon on the OS/2 desktop.

Installing LCU: Next, Helen double-clicks on the CASSETUP icon to load the CASSETUP program. She is now ready to begin the setup process. Because CASSETUP is a LAN CID Utility tool, LCU must be installed before CASSETUP can be fully functional.

Helen fills out some information about where files will be placed, then follows the on-screen instructions to insert a few of the LAN Server 4.0 diskettes or the CD-ROM.

Once LCU is fully installed, Helen reboots the Code Server. She opens the Code Server container and looks at the configuration settings. The defaults for Server name and aliases look OK to her, so she is ready to load software onto the Server.

Loading Installable Software: Helen starts to load installable software images onto the Code Server. She first obtains the installation diskettes (or CD-ROM) for the application to be loaded on the Code Server. Then, in the Applications Folder, she finds the icon for the application (there may be separate icons for diskette and CD-ROM versions of packages). CASSETUP comes with icons for OS/2 2.11, MPTS, LAN Server, and LAN Requester in the Applications folder.

Using familiar Workplace Shell operations, Helen drags the application icon

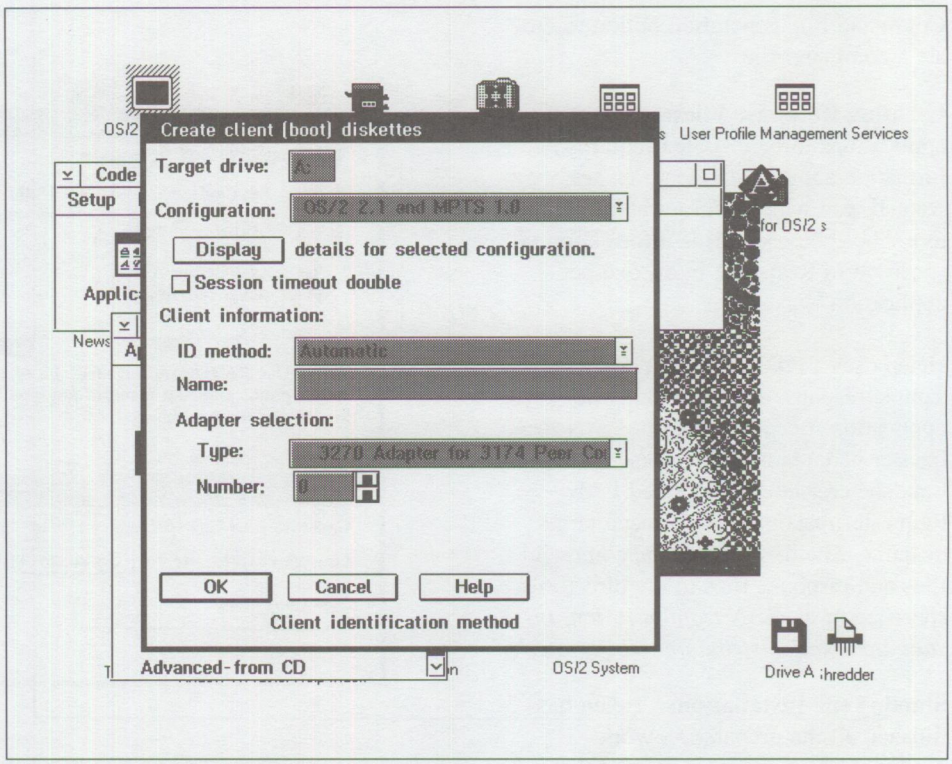


Figure 2. Creating Boot Diskettes

onto the Code Server icon in the Servers folder. Figure 1 shows how applications are dropped onto the Code Server.

After Helen follows a series of on-screen instructions for inserting diskettes, the application is ready to be installed over the network.

Building Boot Diskettes: Helen wants to install OS/2 2.11 on workstations in her network. To do this, she needs to have boot diskettes that contain a bootable OS/2 2.11 operating system.

Helen chooses the "Create client (boot) diskettes" option. Figure 2 shows the dialog she sees. From a list of available boot diskette formats, she chooses the one that will build OS/2 2.11 boot diskettes. After following the on-screen instructions to insert output diskettes, Helen has the required set of boot diskettes. (She may later clone them for use on other workstations, or modify them for a particular workstation, as specified in the *LAN Configuration, Installation, and Distribution Utility Guide.)*

Creating a Command File: Helen wants to install OS/2 2.11 on workstations in her network. She also wants to install Multi-Protocol Transport Services (MPTS) – a separate option when doing LAN CID Utility installations – and LAN Requester on the same machines. To create command files, she selects the "LAN CID Utility CASPREP files ..." option. She sees the dialog shown in Figure 3.

From a list of applications available for installation, Helen selects OS/2 2.11, MPTS, and LAN Requester; then she presses the "Create command file" button. A CASPREP file is automatically generated.

Helen scans the CASPREP file to make sure it contains the right information (using the *LAN Configuration, Installation, and Distribution Utility Guide* as a reference), then uses CASSETUP's LCU

Command File generation option to create a command file.

Creating Response Files: Each of the applications to be installed requires information about which features are wanted, and which configuration parameters should be used. This information is specified in Response Files, one per application.

Helen uses a text editor to modify the sample response files included with each application. Because a unique LAN Requester name is needed for each workstation, she creates a customized LAN Requester response file for each target machine. She then moves the command files and response files to the directories specified in the *LAN Configuration, Installation, and Distribution Utility Guide.*

Starting the Installations: Helen has finished all the preparation work needed. She now transports boot diskettes to the sites where the target workstations are located.

At the target workstation, someone restarts the system from the boot diskettes. The LCU programs on the boot diskettes find the CID Code Server, and the installation of OS/2, MPTS, and LAN Requester proceeds.

Summary: Helen has just accomplished a complex task – loading system software on a workstation located miles from her desk – with very little effort. CASSETUP has given her the tools to handle the difficult and hard-to-remember tasks associated with distributing OS/2 software on her network.

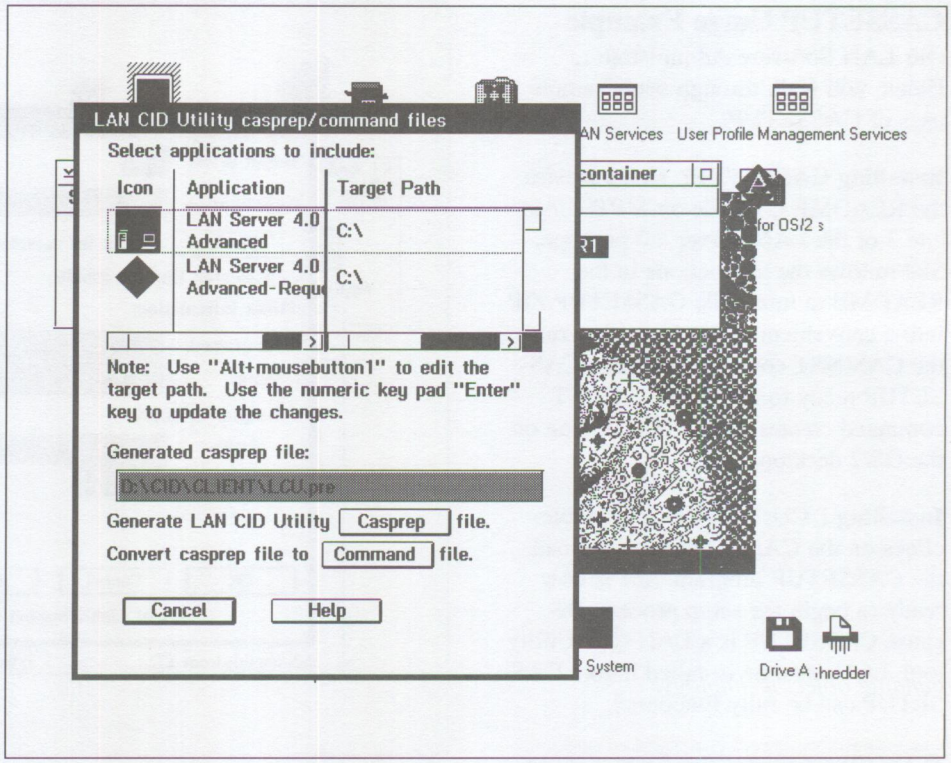


Figure 3. Creating LCU Command Files

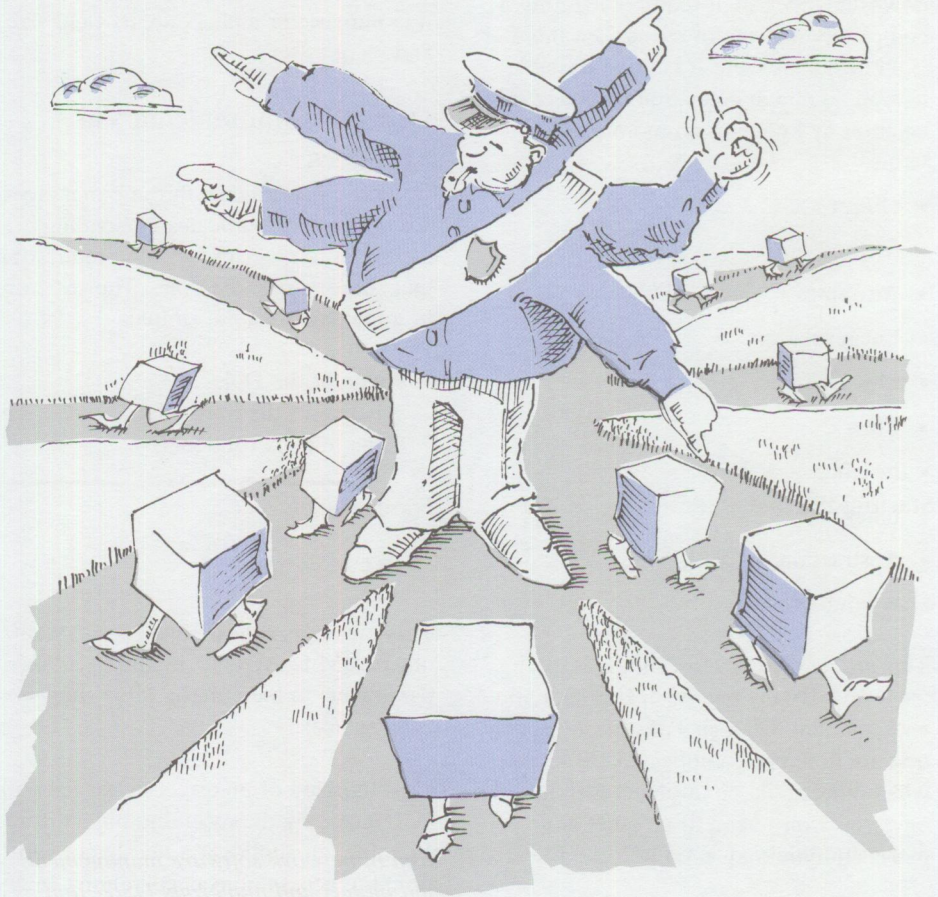
Bob Blair *is an advisory programmer in LAN Systems within the IBM Personal Software Products division in Austin, Texas. He is currently a designer in the area of network software management, and he has had numerous OS/2- and AIX-related design and programming assignments.*

Theodore Shrader *is a senior associate programmer in IBM Personal Systems in Austin, Texas. He began working for IBM in 1989, and his work has included designing and developing graphical, object-oriented interfaces to database and LAN products. He is also co-author of the* OS/2 2.1 Application Programmer's Guide, *published by Van Nostrand Reinhold in 1994 (ISBN 0-442-01736-7 or IBM Order # SR28-5449). This book includes a sample CID installation program written in C. Ted holds a B.S. degree in computer science from the University of Texas at El Paso. (Viva Miners!) He can be reached via Internet at shrader@vnet.ibm.com.*

Object Orientation in Ten Easy Terms

Rainer Laier
E/ME/A Open Enterprise Group
IBM United Kingdom
Basingstoke, UK

Object technology is gaining more and more momentum in almost every area of information technology. This article gives a basic education about the ongoing developments and discussions surrounding object technology.



Why All the Hype About Objects?

Applications are usually written for one specific task, and are therefore called *procedural* applications. Because almost every task is different from other tasks, most of the design and programming work starts from scratch. Even worse, there is little re-use of already available modules of program code. Not only is the development of large systems a complex effort, but maintaining them is also quite a challenge, eating up additional productivity and money.

For all these reasons, developing new applications often takes a long time – too long in today's fast moving and competitive environment. Therefore it is vital that application development must advance quickly from hand-built to automated fabrication, similar to the automobile industry, in which pre-fabricated parts are used to assemble a wide range of cars.

This is what object technology offers. Objects are pre-fabricated parts. Linking together these parts enables programs to be developed in a quicker, simpler way than ever before. Pre-fabrication also equips the programmer with well tested and robust components, a much-valued benefit. Because re-use is one of the cornerstones of object technology, the same objects can be used to build many programs. Combined with more efficient maintenance, application development productivity using object technology is soaring.

Objects are Natural

We live in a world of objects: for example, cars, pencils, and envelopes. All of these objects have *behaviors* and *characteristics*. For instance, the behavior of a a car is to drive, stop, turn, and so on, while a car's characteristics are its speed, its seating capacity, and so on.

In the Information Technology (IT) sense, an object models a real-world object with the help of software. Objects in IT are nothing new; even though many of us have become aware of them in just the last few years, the concept was created in the late 1960s with the advent of Simula, the simulation programming language developed by the University of Oslo.

Ten Terms to Know in Object Orientation

Object orientation, like many other technologies, has developed its own terminology. People might regard this as a downside, because it requires learning a completely new set of terms. But there is an upside as well: it is quite possible to gain a good appreciation of object orientation by knowing no more than ten terms:

- Object
- Method
- Attributes
- Encapsulation
- Message
- Class
- Instance
- Inheritance
- Abstraction
- Polymorphism

This article takes a closer look at these ten terms. By the end of this article, you will find that the basics of object orientation are not very complicated. After that, when others talk about object technology, you won't need to be quiet any more, and you may even become fluent!

Anatomy of an Object

Objects are software packages that contain procedures and data. Procedures that access the data associated with an object are called *methods,* and variables (the data itself) in object terminology are known as *attributes.* Everything that an object knows is expressed in its attributes, and everything that an object can do is expressed in its methods.

Examples of methods are Add, Subtract, and Move. Examples of attributes are Name, Salary, and ID.

Attributes can only be accessed through the object's methods. This enforces correct use of the data.

Object data is *encapsulated* from the outside world. Encapsulation protects the data (the attributes) from being corrupted accidentally. Also, with encapsulation, the system does not need to care about the structure of the data – whether it is numeric or alphabetic, its field size, and so on.

Objects Communicate Via Messages

Objects communicate through messages. An object communicates by sending messages to other objects and requesting that some "work" be done. The structure of a message is quite simple:

Name of the Object
+ Name of the Method
+ Parameters (optional)

For example:

JIM ADD 1000

This message, sent by one object, wants object JIM to ADD (= method) 1000 (= parameter) to the current value (hopefully salary).

Unauthorized or incorrect messages will be rejected by the receiving object; for example, an application that is not supposed to access objects containing salary data will be prevented from gaining access to the data. Only applications that have the correct "entrance ticket" are allowed to access this data.

The combination of encapsulation, methods, attributes, and messages provide the robustness that makes objects less vulnerable to unexpected events.

Classes Bring Order to Objects

Imagine that a firm employs 250 salespeople. For each employee, one object needs to be defined. Using what we have just learned – that every object has its own methods and attributes – in this case the system would have to provide the ADD method 250 times. This would not be very smart in terms of creation and maintenance, and it would consume much system storage as well.

The introduction of the *class* is the way out of this problem. A class is an object that contains all the methods and attributes that are common to similar objects, e.g., objects of salespeople. For example, the class SALESREP might include methods and attributes that are common to all the objects of salespeople. Thus, methods and attributes are defined only once, in the class. This saves the system from the considerable overhead of managing and maintaining 250 copies of the same methods and attributes.

The class concept brings enormous benefits to application development. Because common characteristics are coded only once, adding new objects to a class is no big thing.

An object that belongs to a class is referred to as an *instance* of that class. (In our example, JIM is an instance of the class SALESREP.) Once it is identified as a member of a class, an instance needs to define only its particular values for its attributes. In JIM's case, that value is the amount of his last commission.

Let's continue the example used earlier. A message reaches instance JIM:

JIM ADD 1000

All that the instance JIM does is to look up the method ADD in class SALESREP, and to apply this method to its own value.

Classes are held in *libraries.* Users can buy pre-fabricated class libraries, such as for Graphical User Interfaces (GUIs) and multimedia.

A well-defined OO system can make use of many already available classes. Buying standard classes from specialized vendors might be more economical than developing them, although some of

the time saved must be re-invested in searching through class libraries to find suitable classes to use.

Inheritance

Let's continue the example a bit more. Surely, this company does not employ only salespeople. Methods and attributes used in the SALESREP class can probably be applied to other kinds of employees as well.

One class, the *general* class, encompasses all employees of this company. The general class is also called the *superclass.* In our example, the superclass, labelled EMPLOYEE, contains the methods and attributes of all related classes. Class SALESREP, class ADMIN, and so on are all subclasses of superclass EMPLOYEE.

Subclasses inherit all the definitions of methods and attributes stored in the superclass. The subclasses need only define their own unique characteristics. In object terminology, the extracting of common characteristics is known as *abstraction.* EMPLOYEE is the abstraction of SALESREP; SALESREP is the abstraction of all salespeople.

What happens when an instance receives the message "JIM ADD 1000"? Object JIM, instance of class SALESREP, contacts SALESREP to get the definition of method ADD. But SALESREP does not contain the method, either. Subclass SALESREP needs to ascertain the definition from its superclass EMPLOYEE, where ADD is finally found.

Specialization

In contrast to the relationship of class to instance, a subclass may contain methods and attributes that are special to its superclass. Subclasses can override and add different methods and attributes to the ones that they have already inherited.

For instance, superclass EMPLOYEE does not reflect the flexible payment scheme for salespeople; rather, it covers only the fixed types of salaries. So, a method that is able to calculate flexible payments must be added. Also, the attribute SALARY is different, because a sales rep's salary obviously consists of both fixed and flexible parts. SALESREP therefore overrides the inherited SALARY attribute from its superclass, and adds the necessary method to calculate the variable payment.

Polymorphism

Like class, *polymorphism* is one of the most important characteristics of object orientation. Polymorph is Greek for "many forms."

Polymorphism is a way of hiding different implementations behind a common interface.

Let's return to the employee example. It certainly needs different algorithms to calculate the salaries of the various types of employees. As mentioned earlier, salaries of salespeople are composed of other parts, as are salaries of administrative people. This implies the use of several different messages and method names in order to get the various classes to calculate salaries.

This is where polymorphism can apply its strength. In computer terms, polymorphism is a way of hiding different implementations behind a common interface, i.e., using the same message for initiating operations in different classes.

Defining a unique method name for calculating salaries would allow any salary to be calculated by sending a similar message to all the objects that are concerned with calculating salaries. Examples might be "MANAGER CALCULATESALARY JONES" or "SALESREP CALCULATESALARY". This simplifies programming – the payroll object does not need to know how this method was actually carried out.

Impediments to Object Orientation

A new paradigm like object technology comes with some impediments to its use.

Because object-oriented programming is new, it should come as no surprise that there is a lack of experienced designers and programmers. Companies will have to invest in educating their people, or possibly hiring new university graduates. Not only will this investment cost money, it will also require extensive training in object-oriented programming before any productivity gains are realized.

A paradigm change from procedural to object-oriented programming requires programmers to adopt a completely new way to think and program. Experienced COBOL programmers will find that they are neophytes again. This will certainly generate some resistance to change. Companies migrating to object technology will have to help employees overcome their resistance by giving them encouragement and ample time to get acquainted with the new paradigm.

The two programming languages that application developers are most likely to employ are Smalltalk and C++. While Smalltalk is based solely on object-oriented concepts, C++ is in both worlds – C++ adds to the procedural features of C object-oriented extensions. Every major vendor offers its own C++ compiler. This multitude results in a mess we have known since the days of Babylon: Objects created from one C++ compiler cannot talk to objects created from another C++ compiler. This lack of standards makes interoperability and portability of objects across various plat-

forms very difficult. It is not what users expect from an open system.

Another concern is the immature component market. The limited number of available pre-fabricated classes does not meet all needs. Companies will need to build their own classes. This leads to the biggest challenge: re-use. Classes have to be designed with re-use in mind, which in turn means that objects have to be as generic as possible. Making objects sufficiently generic requires a lot of time, as well as the discipline for taking into account the future re-use of these classes.

All of these impediments slow down the anticipated gain in productivity. In fact, managing an initial object-oriented project is akin to walking a tight rope. On the one hand, the project manager has to convince management that object orientation enhances development productivity, while on the other hand it may require a substantial investment in time. The cost of this investment will make the company's controller unhappy.

But all will turn out well: Object orientation, once implemented, will make the controller smile once again.

Advantages of Object Orientation

Investing the time to develop generic objects will pay off.

Objects can be re-used much more easily than procedural code, so that less new code has to be developed. As a result, development of a finished system will take much less time.

Because object code is used over and over, the same code will become proven, and its quality will be high.

Programs can grow in functionality by linking further objects and classes, so scalability can be achieved more smoothly.

Program maintenance is much easier, because object orientation is the epitome of modularity. Changing a program is much easier, because it is not necessary to rebuild the entire program.

A major plus for object orientation is that it requires and encourages cooperation between programmers and users. Objects are modeled after real-world objects. This relationship makes it easier for users and developers to communicate, which dovetails nicely with the concept of rapid prototyping.

Object technology is maturing rapidly.

Rapid Prototyping

Rapid prototyping is an incremental approach that heavily involves both users and programmers. Under this approach, programmers and users first discuss the functionality necessary to meet business needs. Then the programmers start to build pieces of code from proven objects. This code is the *prototype.* The programmers then show the users what is thus far developed, and the users specify changes. Now, the programmers enhance the prototype – quickly, compared to making changes to procedural programs – and once again they show the prototype to the users. This iterative process continues until the desired result is achieved. No prototype is ever discarded; instead, the iterative prototypes result in the programs that satisfy the user's requirements. The speed attained during each iteration – indeed, the overall speed of the entire project – leads to the term *rapid prototyping.*

As the prototype is gradually being enhanced, with heavy involvement of the user, the danger of misinterpreting what the user wants is minimized. Anyone who has done procedural application development knows that this represents great progress, compared to the usual case where a solution is delivered several months after the initial discussions, while the user was not consulted during the programming.

Object Technology is Inevitable

Object technology is maturing rapidly. Standards are, or are going to be, in place (through the Object Management Group). Application development tools are available, and the initial object-oriented projects have been successful. In fact, because of its potential and advantages, and because it has captured the focus of developers and customers, object technology is inevitable.

Rainer Laier *is project manager of European client/server business in the IBM Europe/Middle East/Africa Open Enterprise Group in Basingstoke, United Kingdom. He was previously a systems engineer for AFP and DB2 in Munich, Germany. Rainer has an engineering degree from FHT University and an economics degree from BA University, both in Mannheim, Germany. His Internet userid is rainerl@vnet.ibm.com.*

IBM United Kingdom
Alencon Link, Basingstoke
Post Box 118, Alencon House
Hampshire RG21 1EJ
United Kingdom

Sharing OS/2 with a PC User Group

E. Gene Barlow
Program Manager, IBM PC User Group Relations
Austin, Texas

If you are like many OS/2 users, you can't imagine why other users have not discovered the wonders of OS/2. After all, it is by far the best PC operating system available today. Chances are that these other users just haven't heard about the wonders of OS/2. Now is your opportunity to share your knowledge of OS/2 with them.

PC user groups are a natural audience for you to preach to about the wonders of OS/2. These user groups are made up of PC industry enthusiasts who have keen interest in the latest hardware and software to hit the market. They are ready and waiting for you to smother them with attention. They will be willing listeners, and eager to try and use OS/2.

Strategy for Working with User Groups

Let's strategize about how you can best work with PC user groups. Armed with a strategy, you will be more effective in how to approach the user groups, and how to service the needs of these groups. By working together, you and PC user groups will mutually benefit.

This article outlines several steps that you can follow.

Special Interest Groups: Once a PC user group reaches a certain size – usually 100 members – the members start to form Special Interest Groups (SIGs) within the overall user group. A SIG meets monthly, and focuses on one specific product of interest to SIG members.

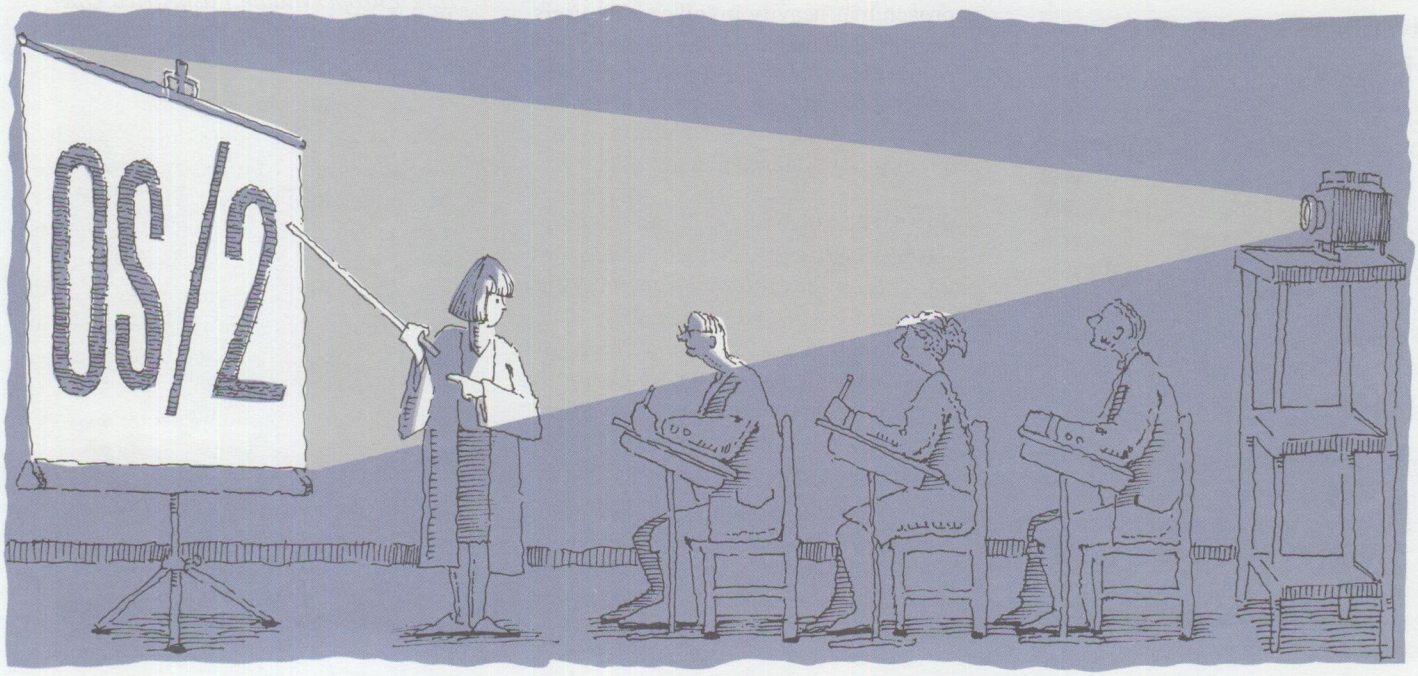
OS/2 is a topic that has spawned SIGs. Many of the larger PC user groups have active OS/2 SIGs, but hundreds of user groups still do not have OS/2 SIGs.

We need to have you join the local PC user groups and to help organize OS/2 SIGs within those groups.

Start an OS/2 SIG: Starting an OS/2 SIG is very easy. All you need to do is to contact the PC user group officer responsible for coordinating SIGs, and volunteer to help him or her get an OS/2 SIG going. The user group's SIG coordinator can probably advise you about how to get the SIG started. It is really as simple as picking a date for the first meeting and finding a meeting location for the SIG (unless the PC user group has facilities you can use). After that, you need to put together an interesting program for the attendees, and to get the word out to the membership.

Once the SIG is started, you need to continue to work with the SIG to help it grow, and to attract more and more members of the PC user group to attend the SIG meetings. This requires work – planning some meetings and presenting fresh topics to the attendees each month – but seeing the interest you will generate in the SIG is well worth the effort.

As soon as the OS/2 SIG is started, make sure you notify IBM's PC User Group Relations team, so we can add your SIG to our database and include



you in future support from IBM. Information for contacting us is given at the end of this article.

The OS/2 SIG is the focal point of all OS/2 activity within the PC user group. So, simply putting on a monthly SIG meeting is not the end of what you can do to assist the local PC user group – it is just the beginning.

Write an OS/2 Article for the User Group Newsletter: Most of the larger PC user groups publish monthly newsletters for their members. A newsletter is the most popular service that a PC user group provides its members, and the only service that reaches 100 percent of the members. So, it is important that a you write an article about OS/2 each month and give it to the user group's newsletter editor to include in the newsletter.

At a minimum, an article should highlight the activities of the OS/2 SIG within the PC user group. Use the newsletter to let the members know which topics you are covering in the SIG meeting, and to pique their interest for next month's meeting. Don't forget to include the date, time, and location of the next SIG meeting, as well as the scheduled topic or guest speaker.

Using the newsletter to promote the next OS/2 SIG meeting is great, but it is not enough. Each month, you should write a short article about how to use an aspect of OS/2. The article should be technical enough to interest readers, but remember to aim it at less experienced users or new users of OS/2. Otherwise, you will not reach the potential new user of OS/2 whom you want to attract to your SIG meeting.

If you feel incapable of writing newsletter articles, assign one of your SIG members to write an article for the newsletter each month. In fact, you might want to assign articles several months in advance, then follow through to make sure they are written as promised.

Another approach worth considering is to look for and read newsletters from other OS/2 user groups; then, in each issue of your own newsletter, you can reprint one of the best articles you find in other newsletters. When you do this, make sure you give proper attribution to the original author of the article, and to his or her user group. Most PC user groups permit other user groups to reprint their articles as long as these credits are given.

One of the best services a PC user group offers its members is to help them get answers to their questions about PC products.

If you don't have access to other user group newsletters, spend some time reading the user group newsletters stored on the IBM BBS in Research Triangle Park, North Carolina. Each OS/2 SIG that registers with us is given toll-free access to the BBS. Remember to upload your OS/2 articles to the BBS so that other OS/2 groups can benefit, just as you may be able to use material from other groups' newsletters.

Put OS/2 Software in Your User Group Library: Another service most PC user groups offer their members is a software library, where members may copy or purchase (for a small fee) software that is in the public domain or designated as Shareware. Because of the interest you are generating for OS/2 in the PC user group, you need to make sure that a selection of OS/2 freeware and shareware is available in your group's software library for interested members. You can obtain much of this software from one of the OS/2 BBSs or from other OS/2 user groups. Just make sure that the software is virus-free and legal to copy.

Start an OS/2 Discussion on Your Group's BBS: PC user groups were among the first to adopt bulletin-board systems. Today, most PC user groups either run their own BBS or patronize a local BBS in their community. Some of the larger PC user groups may offer multiple BBSs to their members for exchanging information and views. These electronic forums are just waiting for you to start a discussion thread about OS/2, and to respond to user questions about OS/2. By monitoring the BBS on a regular basis, you can generate lots of interest in OS/2 within the user group. The BBS is also a place to promote your OS/2 SIG meetings, and to highlight some of the interesting topics brought up at SIG meetings.

Offer an OS/2 Help Line for the User Group: One of the best services a PC user group offers its members is to help them get answers to their questions about PC products. Many groups set up lists of volunteers who are willing to answer technical questions about a variety of hardware and software products. Ask your group if they have such a list of helpful volunteers, and add your name to the list as the OS/2 expert.

You provide a phone number and specify the days and hours during which you are willing to receive phone calls. Most questions you get will probably be very easy for you to answer, since you are an experienced OS/2 user, but to a new user of OS/2 even the basics may be difficult at first. So, lend a technical hand to your fellow user group members, and get them to join your OS/2 SIG. Before long, they will be answering questions from other members!

Arrange an Annual OS/2 Presentation to Your User Group: About once a year, you should work with the program chairman of your PC user group to arrange for an OS/2 presentation to the en-

tire group. This is a way to build the membership of your OS/2 SIG, and to encourage other members to try OS/2. Some OS/2 SIG members feel comfortable presenting OS/2 to the entire group themselves. Others prefer to have an outside speaker come in for this presentation. In either case, your user group will benefit from hearing about the latest version of OS/2.

Try to pick a date for the presentation that is close to a major new release of OS/2. As an OS/2 enthusiast with an ear to the ground, you know when exciting things are about to happen with OS/2. Contact the PC user group's program chairman as soon as you think a new release of OS/2 will be announced. Most of the larger user groups schedule their speakers several months in advance, so getting a date committed to OS/2 early is key to catching the audience at the same time that the new OS/2 release will be receiving lots of trade press and attention. This timing should work in your favor to bring in a larger audience to see and hear about the latest OS/2 features.

As soon as you choose a date, contact IBM's PC User Group Relations team so that we can record your meeting date in our database of upcoming OS/2 presentations. If you don't have a speaker arranged, we can help you to line up a super presenter in your area. If your SIG is planning to present the meeting, we may be able to assist you with the latest information to present. In both cases, we will be able to ship you a box of goodies to use as handouts and door prizes at the meeting. It's amazing how an offer of free software or merchandise will attract and hold an audience!

Once the meeting is scheduled with the user group and with IBM, all that remains is to make sure that the details of the presentation are taken care of. We arrange hundreds of these meetings each year, and can help you anticipate all the things that need your attention. Again, discuss these items with us when you call us to register your event.

Especially important is the amount of promotion you do before the meeting. OS/2 is an interesting topic to many of the group's members, and it usually draws a larger-than-normal crowd to the presentation. But this natural interest won't happen if you don't get the word out about the topic of the meeting and all the planned activities and giveaways. Work with the user group's newsletter editor to run a lead article about OS/2 and all its advantages in the issue that comes out right before the meeting. Find some exciting OS/2 graphics to place on the cover of this newsletter.

If you don't have a speaker arranged, we can help you to line up a super presenter in your area.

Make sure the meeting is announced at the general user group meeting held the month before the OS/2 presentation. Also, announce it at your OS/2 SIG meeting to gain the support of your SIG members. And don't forget to place a notice on the group's BBS to promote the meeting. If you have funds or access to the local press, run an ad or a notice in the local newspaper to try to draw nonmembers from the community to the OS/2 presentation. Many of them have heard about OS/2 in the press and would welcome a chance to see it in action. They may also become interested in joining the user group and your OS/2 SIG.

Another idea is to print simple flyers announcing the OS/2 presentation, and make them available at your local OS/2 dealer. You may even want to invite the dealer to bring in copies of OS/2 to sell at the meeting at a special user-group price. Make sure that the PC user group officers are aware of this plan, and get their blessing in advance. They may have restrictions against such commercial activities at the meeting, and may prefer that the dealer pass out a special discount coupon at the meeting instead. Either way will work!

Participate with Other OS/2 SIGs Around the World: When your PC user group has an active OS/2 SIG program and is servicing your new OS/2 users in the many ways outlined in this article, consider reaching out to other OS/2 user groups and OS/2 SIGs around the world, and learn to work together and exchange ideas with these other groups.

Hundreds of PC user groups belong to the international Association of PC User Groups (APCUG), which is set up to help all user groups – including OS/2 groups – meet and share with each other. Membership in the APCUG costs a user group only $25 a year, and the benefits of membership far outweigh this negligible fee. Your PC user group may already be a member of the APCUG. If it is, look into what services the APCUG offers that your OS/2 SIG can use. If your group does not belong to the APCUG, encourage the group to join – it's well worth it!

In addition, you can take the lead in working with the APCUG to help promote OS/2 across all user groups. That way, the entire OS/2 user group community will come together and become a major force in promoting OS/2 in the industry.

What About Independent OS/2 User Groups? In some communities, independent OS/2 user groups – not connected with PC user groups – are already in place to service the needs of the OS/2 users in the community. If you are a member of an independent OS/2

user group, consider reaching out to the local PC user group in the community and forming an alliance between the two groups. Try to work out a way that the two groups can bind together, so that members in each group can totally share with the other group.

If the two groups can't come together as one, then join the PC user group and help them set up an active OS/2 SIG within the PC user group. Its members need exposure to OS/2 that will best come through the OS/2 SIG. Later, as these OS/2 SIG members gain in their knowledge of OS/2, they will naturally want to join the local OS/2 user group as well as continue with the PC user group. Just make sure that the OS/2 SIG is not depleted by the larger OS/2 user group, which would leave the PC user group without an active OS/2 SIG to service its members.

How Can IBM PC User Group Relations Help You?

When IBM first announced the Personal Computer, a PC User Group Relations team was put in place to service the needs of these important groups. Ever since then, IBM's PC User Group Relations team has continued to work with hundreds of PC user groups around the world.

About three years ago, when it appeared that OS/2 was destined to become a great product, IBM started an active support program for OS/2 user groups. Today, we support over 100 independent OS/2 user groups, and another 100 OS/2 SIGs available to users of OS/2.

That is just the start. We should have many times that number of groups, and with the help of you out there, we can attain that goal.

Contact us to help you find a PC user group in your area that needs an OS/2 SIG. We can also send you some information about how to start an OS/2 user group or SIG, and what sorts of activities they offer their members. As soon as you get your SIG organized, let us know so that we can register you with us for support.

Contact us to help you find a PC user group in your area that needs an OS/2 SIG.

As a supported OS/2 SIG, you will get toll-free access to the IBM BBS in North Carolina. We will also start mailing you a monthly package of OS/2-related information and support items to help your OS/2 SIG. Each quarter, we will send you multiple copies of our *Personal Software* Magazine, which features OS/2 and LAN systems articles from IBM and other OS/2 user groups and SIGs. We will also work with you to arrange for IBM speakers at your annual meetings or SIG meetings and, of course, furnish all the goodies you need to make your meetings a success.

We are also very active in the PC user group community, and are one of the major supporters of cross-user-group officers' meetings and exchanges. We also work with other vendors who support user groups, to help them improve their offerings to groups such as yours. We are active in the exchange of information between OS/2 user groups and PC user groups, and are always looking for ways to improve our service to these groups.

How Can You Contact IBM PC User Group Relations? We can be reached by mail at the following address:

Gene Barlow, Program Manager
IBM PC User Group Relations
P.O. Box 201449
Austin TX 78720-1449

Our e-mail address on Internet is:

ibmpcug@vnet.ibm.com

You can also reach us by phone. Richard Woolsey, our OS/2 SIG coordinator, can be reached at:

1-512-823-1856

Please remember that our user group activities may keep us away from our phones at times, so try to contact us by e-mail for the quickest response.

We hope that this article has given you a new perspective of what you can do to work with PC user groups in promoting OS/2 to the user group community. We are anxious to work with you in making this happen. Please let us know how we can better serve your needs as you work with PC user groups.

***E. Gene Barlow**, Program Manager*
IBM PC User Group Relations
Internet: gbarlow@vnet.ibm.com
Phone: 1-512-823-3259
Fax: 1-512-823-3252 (M-F, 8-5 Central Time)

Trademarks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation:

IBM, AIX, AS/400, Micro Channel, Operating System/2, OS/2, Person to Person/2, PS/2, PowerPC, Presentation Manager, RS/6000, ThinkPad, WebExplorer, WIN-OS/2, and Workplace Shell.

The following terms, denoted by a double asterisk (**), are trademarks of the companies noted:

Advantis – Advantis
Amiga – Commodore Amiga, Inc.
AutoDesk – AutoDesk, Inc.
cc:Mail – cc:Mail, Inc., division of Lotus Development Corporation
CompuServe – CompuServe, Inc.
FaxWorks – SofNet, Inc.
DEC – Digital Equipment Corp.
Gopher – University of Minnesota
HyperACCESS – Hilgrave Inc.
Intel, Pentium and Indeo – Intel Corporation
Kodak – Eastman Kodak Company
Macintosh – Apple Computer, Inc.
Microsoft, MS-DOS, QBasic, Windows and Windows NT – Microsoft Corporation
Norton Utilities – Symantec Corp.
Novell and NetWare – Novell, Inc.
PCMCIA – Personal Computer Memory Card International Association
Quicken for Windows – Intuit Company

All other product names may be trademarks of their respective companies.

“**OS/2 Warp includes a BonusPak of more than ten useful applications.** (page 4)

“**Regardless of the mixture of application types, OS/2 smoothly multitasks dozens of concurrent programs.** (page 13)

“**OS/2 Warp Version 3 has all the “right stuff.”** (page 23)

“**You can modify many aspects of your OS/2 configuration by making changes to the CONFIG.SYS file.** (page 26)

“**A REXX program that installs objects on the desktop can also create the required application directory and file structure.** (page 46)

“**The new LAN Server Administration GUI is based on a design that utilizes objects and object containers.** (page 69)

“**Response files remove the need for the user to become an expert on the application’s installation and configuration options.** (page 83)

“**CASSETUP is a tool for automating and guiding a user through some of the more complex, time-consuming tasks of software management in a network.** (page 85)

“**Object technology is maturing rapidly.** (page 92)